FACULDADE DE ENGENHARIA DA UNIVERSIDADE DO PORTO

Localização de Emissões de Rádio por SDR

Pedro Miguel Marinho Teixeira

Mestrado Integrado em Engenharia Electrotécnica e de Computadores

Orientador: Prof. Dr. Sérgio Reis Cunha

26 de fevereiro de 2016

Resumo

O desenvolvimento da tecnologia de rádio baseado em *software* permite que um único dispositivo receba virtualmente qualquer tipo de sinal, independentemente da sua frequência ou modulação, deixando todo o processamento de sinal encarregue a módulos de *software*. Esta tecnologia está em crescimento e conta com produtos disponíveis no mercado há alguns anos. O maior entrave à aquisição dos dispositivos *Software-defined Radio* (SDR) era o seu elevado preço. No entanto, existem recetores de televisão terrestre (DVB-T) cuja composição, em termos funcionais, permite a sua utilização como dispositivos de *software defined radio*. A descoberta desta funcionalidade despoletou a criação de uma forte comunidade *online* dedicada à experimentação com este tipo de dispositivos, visto que o preço deste tipo de recetor é bastante baixo quando comparado com outros produtos SDR presentes no mercado.

O objetivo desta dissertação segue o espírito de experimentação, pretendendo tomar partido das possibilidades disponibilizadas por este tipo de componente *low-cost* para estimar a posição de emissores. O caso de uso principal consiste na integração do sistema com um automóvel, por forma a detetar a direção de emissões de rádio em movimento.

Para atingir o objetivo estabelecido, o trabalho foi dividido em componentes funcionais por forma a escolher os módulos de *software* e *hardware* que melhor se adequam às necessidades do sistema. Como forma de apurar a escolha, estipulam-se orçamentos totais de consumo de potência e custo monetário a serem respeitados.

Em seguida à escolha de dispositivos, é feita a integração dos componentes funcionais e procede-se com a análise do desempenho do sistema. O desempenho do sistema é analisado através de testes com emissores cuja localização é conhecida, por forma a comparar com os resultados do sistema.

Palavras-chave: Software-defined radio, radio source localization, direction-of-arrival estimation, localização de emissões de rádio

Abstract

The development of radio technology based on software enables the reception of virtually any type of radio signal by a single device, independently of the signal's frequency or modulation, by focusing signal processing on software modules. This technology is currently growing and there have been commercially available products on the market for a few years. The major issue with Software-defined Radio (SDR) devices used to be their market price. However, there are low cost Digital Video Broadcasting - Terrestrial (DVB-T) receivers whose functional structure allow their use as SDR devices. This revelation has triggered the rise of a strong online community dedicated to experimentation with this kind of devices, mainly due to its low price when compared with other SDR products currently available on the market.

The present dissertation's main goal is to explore this low cost SDR receiver in order to deploy it as the radio-frequency front-end of a radio source localization system. The main use case consists on the integration of the developed system with a car in order to detect radio emissions' direction while the vehicle is in motion.

In order to reach the established goal, the project was segmented in several functional components to ensure a selection of the software and hardware modules that would offer a performance compliant with the system requirements. Power and monetary budgets were specified to establish criteria of choice between the functional components.

The integration of the selected functional components is also documented, followed by the system's performance analysis. The performance of the system is analysed through the testing of the system's results when estimating radio sources' localization whose positions are known.

Keywords: Software-defined radio, radio source localization, direction-of-arrival estimation

Agradecimentos

O meu percurso académico não seria possível sem o apoio incondicional da minha mãe, Aurora Marinho, e do meu pai, Miguel Teixeira. Agradeço-lhes acima de tudo por nunca terem desistido de mim. Deixo uma palavra de apreço à pessoa que mais influenciou o meu carácter, o meu irmão Marco Teixeira, que, apesar de estar longe, continua a ser o meu mentor e o meu mais fiel amigo. Ao resto da minha família, um enorme obrigado por me ajudarem a crescer num ambiente saudável e acolhedor. Espero um dia conseguir retribuir tudo o que estas pessoas fizeram por mim.

Quero agradecer à minha companheira, à minha grande amiga, à minha namorada Ana Rebelo, por toda a força e apoio inesgotáveis que sempre me dedicou. Esta vitória também é tua.

Ao Prof. Sérgio dedico todo o meu respeito e consideração pela sua excelente orientação no decorrer deste trabalho. Aprendi imenso sob a sua alçada e foi, sem dúvida, uma das melhores experiências da minha vida. Espero ter absorvido alguma da sua praticalidade e imenso conhecimento desta arte que é a Engenharia Electrotécnica. Quero também agradecer à Zaida Silva pela disponibilidade, pelas palavras sábias e pela motivação. Esta vivência não teria sido a mesma sem a sua presença.

Presto o meu agradecimento ao técnico Vítor Pinto por todo o trabalho que fez no âmbito desta dissertação e pela sua prontidão em disponibilizar ajuda para o que quer que fosse. Agradeço também ao técnico Pedro Alves pelas dicas que me forneceu sobre as técnicas de desenho de circuitos e pelo seu profissionalismo. Dedico um sincero obrigado ao Prof. Henrique Salgado por ter conseguido motivar o meu interesse pela área da Tecnologia das Comunicações e por me ter ajudado sempre que eu precisei.

Aos meus colegas Rui Feio, João Machado, Bruno Vieira, Tiago Costa, Erick Lima Delgado, Rui Gomes e Diogo Lage, agradeço a companhia, a ajuda e as conversas que tivemos ao longo deste trabalho. Muito obrigado.

Por último, faço questão de dedicar este trabalho à minha avó Maria Fernanda Cunha que, infelizmente, já não se encontra entre nós. Serve esta dedicatória para agradecer as noções de perseverança, força de vontade e sentido de comunidade que esta mulher gravou na minha personalidade. À tua memória.

Pedro Marinho

"Hey – don't worry, don't be afraid, ever, because, this is just a ride..."

Bill Hicks

Conteúdo

Lista de Figuras

Lista de Tabelas

Abreviaturas e Símbolos

ADC	Analog-to-Digital Converter
AM	Amplitude Modulation
AOA	Angle-of-Arrival
ARM	Advanced RISC Machine
AWGN	Additive White Gaussian Noise
CMOS	Complementary Metal-oxide-semiconductor
CPU	Central Processing Unit
DAB	Digital Audio Broadcast
DDC	Digital Downconverter
DSP	Digital Signal Processing
DVB-T	Digital Video Broadcasting - Terrestrial
EDA	Electronic Design Automation
FIR	Finite Impulse Response
FM	Frequency Modulation
GPIO	General Purpose Input Output
IDE	Integrated Development Environment
IF	Intermediate Frequency
LDO	Low-dropout Linear Regulator
LFCSP	Leadframe Chip Scale Package
ML	Maximum Likelihood
PCB	Printed Circuit Board
PRBS	Pseudorandom Bit Sequence
RF	Radiofrequência
RISC	Reduced Instruction Set Computer
RSS	Received Signal Strength
SDR	Software Defined Radio
SNR	Signal-to-Noise Ratio
TCP/IP	Transport Control Protocol / Internet Protocol
TDOA	Time Difference Of Arrival
TOA	Time Of Arrival
TSSOP	Thin-shrink Small Outline Package
USB	Universal Serial Bus
USD	United States Dollar

Capítulo 1

Introdução

A localização de emissores rádio é um dos aspetos de maior utilidade para um número de áreas cuja base de comunicação é a emissão de sinais de rádio-frequência (RF). Estas áreas vão desde as comunicações por satélite até à vigilância militar aérea e ao radioamadorismo.

O desenvolvimento de equipamentos *low cost* com arquitetura *software defined radio* (SDR) tem vindo a estabelecer uma base de investigação incidente sobre os potenciais usos para este tipo de equipamento, para além da comunicação ponto-a-ponto oferecida a nível comercial. A presente investigação tem como objetivo a produção de uma plataforma para localizar emissores de rádio de forma automática usando SDR.

1.1 Motivação

O avanço da tecnologia RF para consumidores comuns permitiu o desenvolvimento do paradigma SDR em equipamentos de muito baixo custo, primariamente utilizados para outros propósitos. A aquisição de um sistema SDR versátil dentro de uma banda de frequências extensa revela um grande número de aplicações.

No contexto desta dissertação, a motivação é conseguir localizar emissores a partir da receção e processamento de sinais, utilizando equipamento SDR para esse fim. Aliar as técnicas de localização de emissores rádio à receção RF do equipamento SDR faz com que o processamento de sinal, isto é, as estimativas da posição e direção de emissor sejam calculadas por um computador, tornando estas estimativas, em princípio, mais fiáveis.

1.2 Objetivo

O objetivo primário do sistema a conceber é a estimação da posição de emissores usando equipamento SDR. Para chegar a este objetivo, será usado o conhecido *chip* RTL2832u integrado dentro de um *dongle USB* disponível no mercado.

Para realizar a receção dos sinais, será concebida uma plataforma de receção multi-antena com um sistema de comutação controlado por um microprocessador, por forma a obter amostras do sinal de interesse a instantes de tempo diferentes a fim de computar a posição do emissor. O sistema desenvolvido terá como caso de uso principal a sua integração no tejadilho de um automóvel, para que a deteção da direção de emissões rádio seja feita em movimento.

Outro objetivo deste projeto é a integração com vários tipos de emissores, nomeadamente a integração com o programa STRAPLEX [1], a integração com o programa GAMASAT [2] e uma possível análise académica de comunicações na área da aviação.

Capítulo 2

Revisão Bibliográfica

Por forma a contextualizar o projeto proposto pela presente dissertação, revela-se oportuno elaborar uma revisão geral das técnicas empregues na localização de emissores no contexto da radiofrequência, assim como fornecer uma visão global sobre os equipamentos baseados na arquitetura *software defined radio*.

2.1 Técnicas de Localização de Emissores

A literatura documenta um conjunto significativo de técnicas de localização de emissores, sendo que a motivação para o desenvolvimento deste tipo de técnicas emergiu de forma quase simultânea à utilização do espetro eletromagnético como meio de transmissão de informação. Na presente secção são apresentadas as técnicas de localização de emissores clássicas e outras desenvolvidas mais recentemente. O conjunto de técnicas apresentadas seguem a denominação *direction finding* presente na literatura da área. Para implementar este tipo de técnicas, pressupõe-se a utilização de *arrays* de antenas.

2.1.1 AOA

As técnicas baseadas em ângulos, conhecidas como técnicas *Angle of Arrival*, fazem uma estimativa da posição do emissor ao medir o ângulo de chegada à estação que faz as medições. O emissor situa-se na linha reta formada pela estação de medição e o AOA estimado, numa quantidade chamada *Line of Bearing* (LOB).

Para calcular o ângulo de chegada de um sinal incidente é utilizada um *array* de antenas direcionais, com, por exemplo, dois ou mais elementos de corrente. O método básico é medir a diferença de fase entre os sinais aquando da incidência nos diferentes elementos de corrente e fazer a conversão para uma estimativa AOA. Este método é conhecido como inferometria e está representado na Fig. 2.1.

Existe outro método AOA conhecido como *beamforming*. Este método consiste na utilização de cortinas de antenas conhecidas como *smart antennas* onde são implementadas técnicas de *beamforming* de forma a direcionar o lóbulo principal de radiação na direção de chegada do sinal

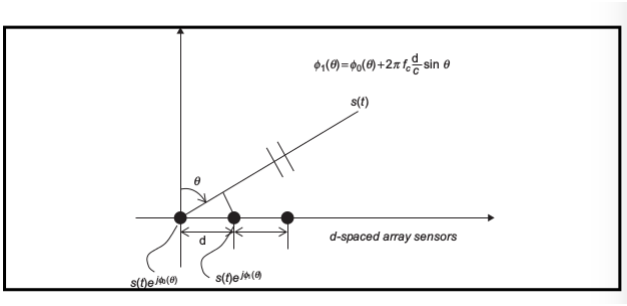

Figura 2.1: Interferometria AOA

pretendido. Existem essencialmente três categorias de técnicas AOA que usam a capacidade de filtragem espacial dos *arrays* de antenas:

- Técnicas convencionais
- Técnicas baseadas em subespaços
- Técnicas de máxima verosimilhança

Os *métodos convencionais* baseiam-se nas técnicas clássicas de *beamforming* e não exploram qualquer previsão sobre o modelo do sinal recebido ou do ruído. A estimação AOA convencional é feita através de um varrimento controlado eletronicamente do lóbulo principal de radiação para detetar picos de potência. A desvantagem deste tipo de métodos é o facto da resolução angular estar limitada pela largura de feixe da cortina de antenas, logo, quanto menor o número de elementos de corrente, pior será a resolução.

Os *métodos baseados em subespaços* são técnicas sub-ótimas de alta resolução que exploram a estrutura de valores próprios dos dados recebidos quando estes são organizados em matrizes tempo-espaço. O algoritmo mais conhecido é o MUSIC (*multiple signal classification*).

As *técnicas de máxima verosimilhança* são técnicas ótimas que ultrapassam a prestação dos métodos de subespaços até em caso de baixos valores SNR. A desvantagem é serem computacionalmente exigentes.

As principais desvantagens das técnicas AOA são as necessidades de calibração periódica dos *arrays* de antenas e de utilização de hardware complexo. As técnicas AOA ficam aquém das expetativas quando comparadas com técnicas TDOA ou TOA em aplicações práticas, uma vez que a estimativa de posição é degradada à medida que o emissor se afasta das estações de medição [3].

2.1.2 RSS

As medições *Received Signal Strength* (RSS) separam-se em três técnicas: *power-based ranging*, *fingerprinting* e inferométrica.

Power-based Ranging

A medição da potência de sinal recebida (RSS) é quase sempre possível em meios sem fios. Partindo do princípio de que quanto mais longe se estiver de um determinado ponto de emissão, mais fraco será o sinal recebido, é possível estimar a distância entre dois pontos através da medição do RSS. No sentido de implementar este tipo de técnica é necessário o conhecimento da potência de sinal transmitida inicialmente para conseguir computar a diferença entre a potência transmitida e a potência de sinal recebido e obter uma estimativa de distância. A vantagem deste método é o facto de não ser necessário qualquer tipo de sincronização temporal. No entanto, os fenómenos de propagação de sinal tais como a refração, a reflexão e o *multipath* fazem com que ocorram atenuações que têm baixa correlação com a distância fazendo com que a estimativa não seja precisa.

Fingerprinting

O método *fingerprinting*, também conhecido como *mapping* ou *scene analysis*, é um método de mapeamento de medições de dados feito num ponto intermédio conhecido que representará uma "impressão digital" dos dados. Esta impressão digital é conseguida através de uma série de testes feitos ao ambiente de medição durante a fase de calibração do sistema. Durante o funcionamento do sistema, os dados recebidos são comparados com as impressões digitais existentes. A grande desvantagem é a sensibilidade das medições a alterações de geometria, como por exemplo o fechar de portas.

Inferométrico

Esta técnica consiste na transmissão de ondas sinusoidais a frequências ligeiramente diferentes por parte de dois pontos distintos. A envolvente do sinal composto recebido varia lentamente com o tempo. O deslocamento de fase da envolvente pode ser estimado através de medições de RSS e contém informação sobre a diferença de distância entre os pontos emissores.

2.1.3 TOA

Consultando o livro *Satellite and Terrestrial Radio Positioning Techniques, A Signal Processing Perspective* [3], pode constatar-se o seguinte: a velocidade de ondas eletromagnéticas a propagarem-se no vácuo é constante ($c = 3 \cdot 10^8\ m/s$) e a partir dessa informação é possível calcular a distância entre um par de nós A e B através da medição do atraso de propagação, ou tempo de vôo, ou *time of flight*, $\tau_f = \frac{d}{c}$, onde d representa a distância entre A e B.

No esquema *one-way ranging*, o nó A envia um pacote de dados no instante de tempo t_1 ao nó B. O pacote contém o *timestamp* t_1 no qual a transmissão foi iniciada. O nó B recebe o pacote no instante de tempo t_2. No caso de os nós estarem sincronizados em relação ao mesmo relógio de referência, o atraso de propagação será calculado como $\tau_f = t_2 - t_1$ e a distância é facilmente calculada. No entanto, se houver erros de sincronização, as medições de distância são severamente afetadas.

No esquema *two-way ranging* ou *two-way TOA*, o nó A envia um pacote para o nó B, que, por sua vez, emite um pacote de *acknowledgment* para o nó A após um determinado tempo fixo τ_d. O

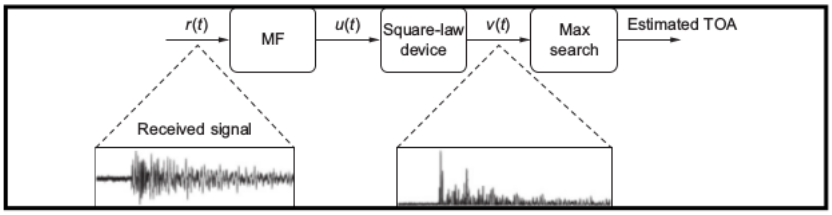

Figura 2.2: TOA

round-trip time (RTT) entre o instante de transmissão do nó A e o instante de resposta recebida é

$$RTT = 2\tau_f + \tau_d, \tag{2.1}$$

de onde é possível calcular a distância assumindo que τ_d é um valor conhecido. O erro devido à sincronização imperfeita é eliminado, no entanto, possíveis variações do relógio podem afetar a estimação de distância.

Para investigar o método de estimação TOA clássico, considere-se a situação onde um sinal genérico $g(t)$ com duração T_g é transmitido e recebido sem distorção com uma determinada energia E_g, assumindo um canal com ruído branco gaussiano aditivo (AWGN). O sinal recebido pode ser modelado da seguinte forma,

$$r(t) = \sqrt[2]{E_g} \cdot g(t-\tau) + n(t) \tag{2.2}$$

onde $n(t)$ representa o ruído AWGN com valor médio em zero e densidade espetral de potência $N_0/2$ e onde τ representa o parâmetro TOA a ser estimado a partir de $r(t)$ num período de observação $[0, Tb]$. Numa primeira fase, o sinal $r(t)$ passa por um filtro adaptado a $g(t)$ visto na Fig 2.2. O filtro adaptado consiste num bloco que realiza a correlação entre um sinal conhecido (*template*) e um sinal capturado, sendo que este processo quantifica a semelhança entre o sinal conhecido e o sinal capturado. A estimativa TOA é calculada através da observação do instante correspondente ao pico máximo à saída do filtro adaptado, ao longo do período de observação. Este tipo de esquema pressupõe uma estimativa de máxima verosimilhança, ou *maximum likelihood* (ML), que garante um estimador TOA $\hat{\tau}_c$ não enviesado para uma relação sinal-ruído de valor elevado e cuja variância é igual a

$$\sigma^2_{\hat{\tau}} = \frac{1}{(8 \times \pi^2 \times \eta \times \beta^2)}, \tag{2.3}$$

onde η representa a relação sinal-ruído $\frac{E_g}{N_0}$ e o parâmetro β^2 representa o segundo momento estatístico do espetro $G(f)$ de $g(t)$ definido como

$$\beta^2 = \int_{-\infty}^{+\infty} \frac{f^2 |G(f)|^2}{E_g} df \tag{2.4}$$

A variância $\sigma^2_{\hat{\tau}}$ representa o limite inferior de *Cramér-Rao*, $k = \sigma^2_{\hat{\tau}}$, que calcula o limite inferior da estimativa do erro quadrático médio de qualquer estimador não enviesado $\hat{\tau}$ de τ, ou seja,

$$Var(\hat{\tau}) = E[(\hat{\tau} - \tau)^2] \geq k \tag{2.5}$$

O denominador da variância do estimador $\hat{\tau}$ é proporcional à energia no sinal, onde a constante de proporcionalidade β^2 depende da forma do impulso. Ao contrário dos métodos baseados em potência, a capacidade de cálculo da distância depende da estrutura do sinal. Para elevados valores de β^2, ou seja, para um sinal com largura de banda extensa, o cálculo da distância será mais preciso.

2.1.4 TDOA

Segundo [3], os sistemas baseados em *time difference of arrival* (TDOA) calculam a diferença de distância entre recetor e emissor. A estimativa TDOA define hiperbolóides de diferenças de distâncias constantes, com pontos de âncora nos focos geométricos. A medição deste tipo de parâmetro ultrapassa as condições necessárias para efetuar uma localização TOA, nomeadamente a necessidade de sincronização entre estações base e estações emissoras e o requisito de alguma forma de *timestamping* presente no sinal, e fá-lo ao examinar a diferença entre as estimativas TOA do sinal em duas estações base diferentes, ao invés de observar o tempo de chegada absoluto. Um método para computar essa diferença temporal é obter o resultado da correlação cruzada dos sinais incidentes no par de estações base. Tendo um par de estações base BS_1 e BS_2, que recebem os sinais $r_1(t)$ e $r_2(t)$, respetivamente, pode-se calcular a correlação cruzada entre ambos os sinais através de

$$R_{1,2} = \frac{1}{T} \int_0^T r1(t) \cdot r2(t+\tau)dt \tag{2.6}$$

onde T representa o período de observação. No caso de não haver erros relativos a atrasos de propagação, $R_{1,2}$ mostrará um pico para τ igual à medida TDOA.

2.1.4.1 *Short Baseline TDOA*

Em [4] é sugerida a técnica *Short Baseline TDOA*, também conhecida como *Differential Time Difference of Arrival*, onde a diferença temporal entre o sinal que chega a uma antena e o sinal que chega a uma outra antena é usada como uma medida da diferença de caminho. Uma implementação simples desta técnica requer uma antena central que funciona como ponto de referência para comparação do tempo de chegada do sinal, assim como um *array* circular de antenas equidistantes da referência. Se o sinal chegar antes, ao mesmo tempo ou depois do sinal recebido no elemento de referência, a direção de chegada do sinal pode ser calculada, sendo que o sinal recebido mais cedo é associado ao caminho direto do sinal. Este tipo de sistemas requerem medições de atrasos temporais precisas em relação ao tempo de subida ou pico de um impulso de sinal.

2.1.4.2 *Long Baseline TDOA*

Em [4], é introduzido o conceito de *Long Baseline TDOA*, também conhecido como *Range-Difference*. Esta técnica baseia-se no estabelecimento de três plataformas de medição em operação

simultânea com ligações de banda larga de alta qualidade entre elas. Com esta disposição de recursos, apenas um canal de receção por plataforma é necessário. No caso da deteção de ondas contínuas, são necessárias técnicas de correlação para estabelecer uma estimativa de diferença temporal, ao contrário das ondas pulsatórias. A correlação não é tão precisa quanto a deteção de impulsos, no entanto, com uma separação física razoável entre os sensores, a estimativa TDOA pode atingir bons resultados para sinais contínuos.

2.1.4.3 TDOAA

O método *Time Difference of Arrival Averaging* foi proposto por Ralph O. Schmidt em 1972 e generalizado em 1996 [5], [6] e sugere que a soma das diferenças de distância numa malha fechada deve ser zero quando não existe ruído de medição. Aplicando esta técnica nas diferenças temporais estimadas, o erro de medição decresce, fazendo aumentar a precisão das estimativas de localização. Quando a combinação RSS e TDOAA é utilizada, é implementado um mecanismo de correção de erros [7].

2.1.5 Localização por Conjugação de Atrasos

Segundo [8], o método de localização por conjugação de atrasos é uma técnica goniométrica baseada na conjugação de atrasos de propagação. Quando um evento se proporciona a partir do emissor, num dado instante de tempo t_0, a frente de onda propaga-se até aos recetores que, por sua vez, tomam nota dos sinais correspondentes ao sinal transmitido atrasados pelos tempos de propagação. É necessário um circuito que faça a medição dos instantes de chegada do sinal em cada antena em relação a um relógio interno. Estas medições são processadas por um microprocessador que conjuga os atrasos e determina a direção e a distância do sinal recebido.

2.1.6 Antena Direcional Rotativa

O método de *direction-finding* usando uma antena direcional rotativa faz uso do facto do padrão polar de ganho variar em função do ângulo. A direção de chegada pode ser estimada ao captar o sinal de interesse ao longo da rotação da estrutura de receção e fazendo a medição das amplitudes de sinal relativamente à orientação da antena, construindo um histórico temporal por forma a comparar para que direção a amplitude do sinal é maior. É possível fazer boas estimativas no caso das antenas serem bastante direcionais, isto é, se as antenas tiverem uma pequena largura de feixe.

2.2 *Software Defined Radio*

Software defined radio representa um paradigma de *design* para dispositivos de comunicação sem fios. Este termo é usado para descrever um dispositivo que é essencialmente descrito, em termos funcionais, através de *software*.

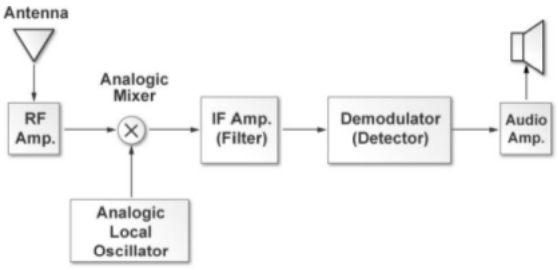

Figura 2.3: Recetor superheterodino

O primeiro dispositivo SDR operacional foi desenvolvido pela Marinha Norte-americana entre 1991 e 1995 [9] e era conhecido como *Speakeasy*. No entanto, o espaço que o dispositivo ocupava e o facto de a aplicação estar dependente do *hardware* revelaram-se desvantagens fatais. A iteração seguinte do equipamento, conhecido como *Speakeasy II* [10], revelou um sucesso muito maior devido aos avanços da tecnologia de comunicação.

Hoje em dia, o *software* e *hardware* para SDR estão disponíveis comercialmente a preços acessíveis ao consumidor comum. A grande parte das implementações de *software* SDR são disponibilizadas *online* sem qualquer custo [10].

2.2.1 Arquitetura

Por forma a compreender a diferença entre o *hardware* de um recetor tradicional de comunicação e aquele que representa um dispositivo SDR, é pertinente estabelecer essa distinção para ver de que formas poderá ser aproveitado o equipamento SDR no ramo das comunicações.

2.2.1.1 Recetor Tradicional

Um recetor tradicional executa três funções fundamentais: sintonização de frequência da portadora para a frequência desejada, filtragem dos sinais irrelevantes e amplificação para compensar as perdas de transmissão. Centre-se a discussão na topologia do recetor superheterodino como exemplo, com o intuito de perceber para que servem os blocos fundamentais que compõem este tipo de recetores. Uma representação dos blocos constituintes de um recetor superheterodino é demonstrada na Fig 2.3.

Tendo o esquema como referência, estabelece-se o significado de cada um dos blocos. Após a captação de um sinal por parte da antena, o sinal é amplificado por um amplificador RF. Por conseguinte, o sinal é alimentado a uma misturadora (*mixer*) que, por sua vez, faz a tradução do sinal para frequência intermédia (IF) através da combinação do sinal recebido com o contributo de um oscilador local. A frequência do oscilador local é definida para um valor tal que a diferença com a frequência do sinal recebido seja igual à frequência intermédia pretendida.

O bloco seguinte é um filtro passa-banda que atenua qualquer sinal fora da sua banda de passagem. A largura de banda do filtro limita a largura de banda do sinal recebido.

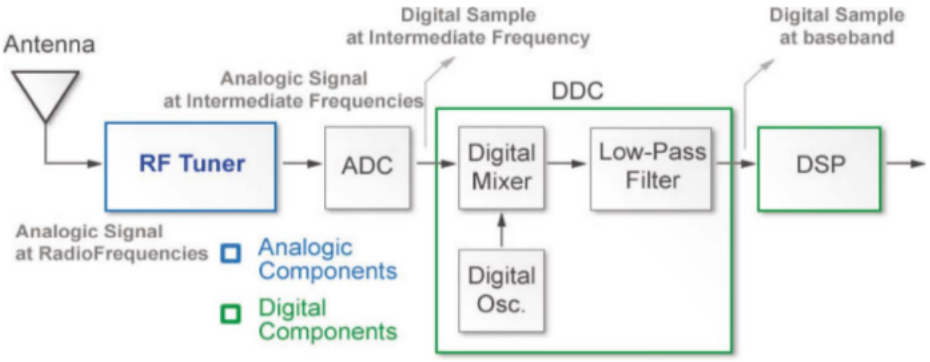

Figura 2.4: Recetor SDR

Por fim, o desmodulador recupera o sinal original a partir do sinal que sai do amplificador IF. Este processo é feito de acordo com a modulação presente no sinal. Para diferentes modulações, há diferentes processos de desmodulação. Por exemplo, em AM, a desmodulação é feita através da deteção de envolvente, enquanto que em FM é necessário fazer a discriminação de frequências. Pode ser necessário o processamento de sinal subsequente, dependendo do propósito do sinal.

2.2.1.2 Recetor SDR

O recetor SDR é representado na Fig 2.4. Numa fase inicial, o sintonizador RF faz a conversão do sinal para IF, fazendo então as três operações essenciais do recetor tradicional. De seguida, o sinal é amostrado por um ADC e as amostras são passadas a um conjunto de blocos conhecido como *Digital Downconverter* (DDC).

Os componentes que compõem este conjunto DDC são: uma misturadora digital, um oscilador local digital e um filtro FIR passa-baixo. A misturadora e o oscilador local servem para traduzir as amostras IF para banda-base, enquanto que o filtro passa-baixo limita a largura de banda do sinal. Os sinais são transferidos para a banda-base equivalente, aquando da sua saída da misturadora digital, por desintegração dos componentes I e Q [11]. No caso da sintonização no oscilador local ser alterada, o sinal pode aproximar-se dos 0 Hz e, em conjunto com a largura de banda do filtro passa-baixo, o sinal pode ser tratado de acordo com a sua utilidade.

Por fim ocorre um processo conhecido como decimação no bloco DSP. Este processo reduz a frequência de amostragem das amostras. Portanto, a frequência de amostragem das amostras em banda-base à saída do recetor SDR é a frequência de amostragem original a dividir pelo fator de decimação N. Também neste bloco são feitas a desmodulação e possível descodificação das amostras.

2.2.2 Equipamentos *Low Cost*

Nos dias que correm, é possível comprar um equipamento SDR a um baixo custo. Este facto abre portas a um infindável número de experiências que podem ser feitas tanto por parte de radi-

Tabela 2.1: Comparação de equipamentos SDR

Designação comercial	Freq. mín. (MHz)	Freq. máx. (MHz)	Largura de Banda (MHz)	Resolução ADC	Transmissão	Preço (USD)
RTL-SDR 2831	24	1766	3.2	8	Não	10-20
Funcube Pro	64	1700	0.096	16	Não	150
Funcube Pro+	410	2050	0.192	16	Não	200
HackRF	30	600	20	8	Sim	300
BladeRF	300	3800	40	12	Sim	400-650
USRP1	10	6000	64	12	Sim	700
MatchStiq	300	3800	28	12	Sim	4500

oamadores como para fins académicos. Na tabela seguinte, existe uma comparação entre alguns equipamentos presentes no mercado [10].

2.2.2.1 RTL2832u

Em 2012, Antti Palosaari descobriu que o *chip* da *Realtek* RTL2832u, que servia como desmodulador em *dongles* USB para televisão europeia, conseguia exportar um fluxo de amostras digitais à saída que descreviam a amplitude e fase (dados I e Q) de sinais dentro de um elevado espetro de frequências [12].

O *chip* RTL2832 é um desmodulador DVB-T de alta *performance* que suporta a utilização de sintonizadores RF a frequências IF (36.125 MHz), *low*-IF (4.57 MHz) e *zero*-IF usando um cristal a 28.8 MHz. Este *chip* contém um ADC de 7 bits para medição de sinais RF, e como este ADC é de boa qualidade, o *chip* tem uma receção estável mesmo em situações de mobilidade elevada [13].

Usando o *chip* RTL2832u, é possível usar ferramentas *open-source* para traduzir informação I/Q em áudio ou fluxos de dados. Com este tipo de equipamento pode-se estabelecer um dispositivo SDR de baixo custo onde as aplicações são variadas: rádio FM, pacotes de dados vindos de *transponders* de aviões e rádio amador [12].

2.3 Conclusões

A revisão dos principais conceitos dentro da área da localização rádio de emissores e da introdução ao paradigma SDR oferece uma visão geral daquilo que é necessário no intuito de construir uma estratégia eficiente para abordar o problema proposto pela presente dissertação.

Figura 2.5: *Dongle USB* baseado no *chip* RTL2832u

No âmbito do projeto em desenvolvimento, será utilizado um *dongle* com o *chip* apresentado em 2.2.2.1 como *front-end* de receção de emissões rádio. A técnica de cálculo de direção de emissão será baseada na técnica *Time Difference of Arrival*.

Capítulo 3

Projeto de Sistema

Este capítulo introduz o sistema projetado para alcançar o objetivo de conseguir localizar emissores através do uso de um *dongle* de receção de televisão digital baseado no *chip* RTL2832u como plataforma *software-defined radio*. Numa primeira fase é apresentada uma visão de alto nível do sistema. Em seguida, as escolhas para os componentes de *hardware* e *software* são justificadas e caracterizadas de forma detalhada. Na secção a seguir, são expostas as bases de processamento de sinal a utilizar na estimação de posição de emissores.

3.1 Visão geral

O objetivo principal deste trabalho é identificar a localização de emissores usando uma *pen USB* com capacidade de captação, amostragem e quantização de sinais contínuos, permitindo um posterior tratamento dos sinais em formato digital. Esta é a premissa básica para o paradigma *software-defined radio*. Sendo que um dos objetivos do trabalho proposto é a utilização deste mesmo paradigma, a plataforma SDR será o bloco central do *design* do sistema. O sistema projetado pode ser representado em alto nível de complexidade através do diagrama de blocos presente na figura 3.1.

Vejamos a função de cada bloco do diagrama 3.1:

Antenas 1-4 Sensores de ondas eletromagnéticas. Apenas uma antena está ativa em cada instante temporal de funcionamento do sistema.

Comutador de antenas 4:1 Bloco que possibilita a escolha entre cada uma das antenas presentes no agrupamento através da comutação entre elas. A taxa de comutação deverá assumir um valor que ultrapasse a largura de banda do sinal recebido, de forma a que não ocorram interferências dentro da banda do sinal.

Microcontrolador Dispositivo de controlo da comutação entre antenas.

Dongle RTL Bloco responsável pela transposição de frequência para banda-base, amostragem e quantização dos sinais capturadas.

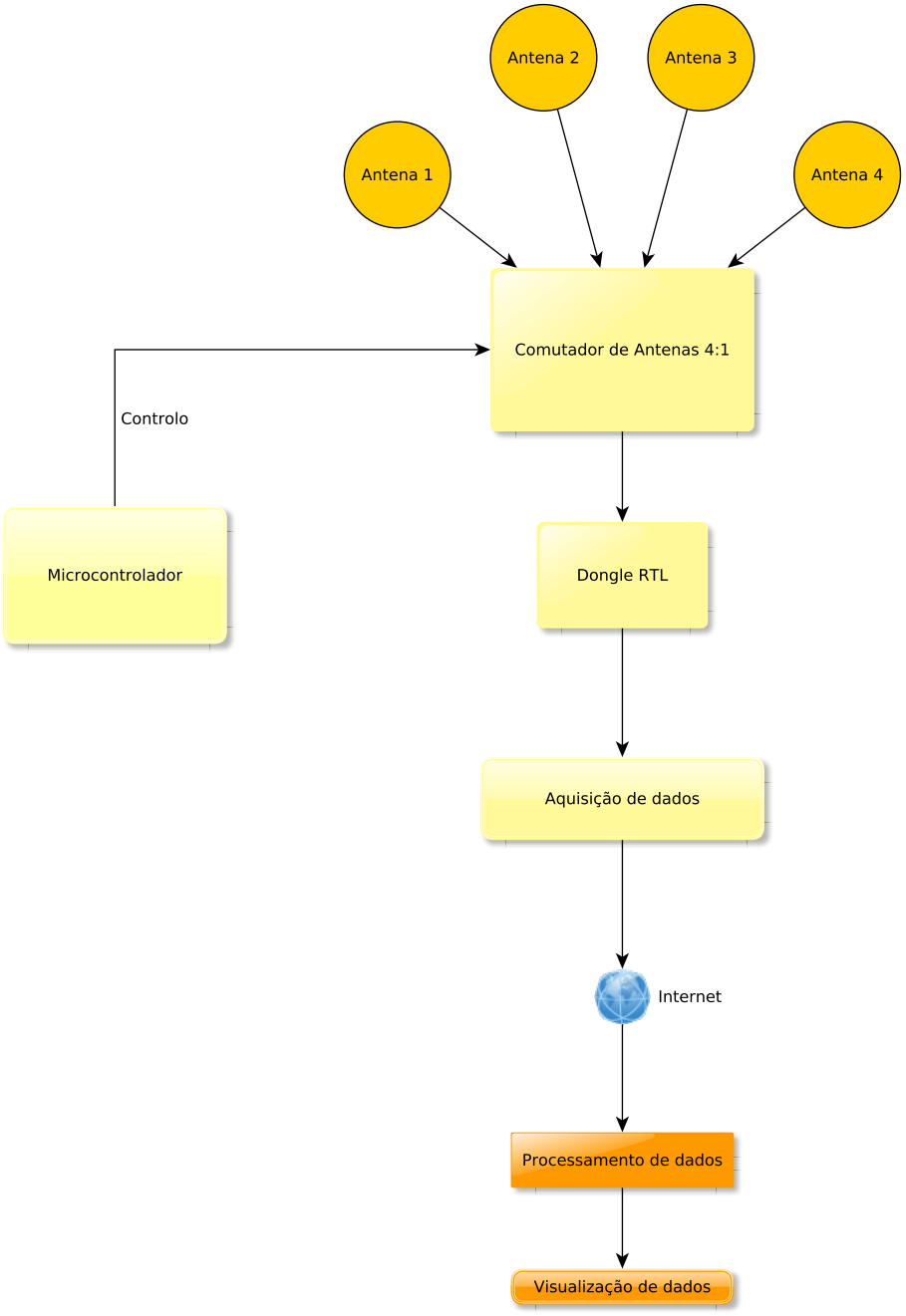

Figura 3.1: Esquema de alto nível do sistema projetado

Aquisição de dados Unidade que recebe e guarda em memória os dados depositados pela unidade USB.

Processamento de dados Nesta etapa, é efetuado o processamento digital dos dados recolhidos.

Visualização de dados Este bloco oferece a *user interface* necessária para a visualização dos resultados obtidos pela aplicação dos algoritmos de processamento de sinal.

Cada ação presente nos blocos do diagrama 3.1 será desempenhada por um determinado componente ou conjunto de componentes de *hardware* e *software*. As escolhas destes componentes serão apresentadas na próxima secção.

3.2 Componentes de *Hardware* e *Software*

Os componentes selecionados para desempenhar as funções descritas pelos blocos do diagrama 3.1 vão de encontro a um conjunto de critérios de escolha previamente definido. Esses critérios são:

- Conformidade com os requisitos do sistema
- Consumo de potência
- Facilidade de integração
- Custo
- Disponibilidade

Os *requisitos do sistema* que os dispositivos escolhidos devem obedecer são os seguintes:

Gama de receção de frequências 24 MHz até 1766 MHz, conforme as características do equipamento RTL-SDR 2831 presente na tabela comparativa 2.1.

Configuração acessível Acessibilidade de programação e disponibilidade de documentação (*datasheets*, *wikis* ou fóruns *online*).

O baixo *consumo de potência* é um aspeto fulcral na concepção deste sistema, visto que assegura a a sua portabilidade. Daí que seja estabelecido um máximo de 3 A de corrente para a totalidade do sistema. Se o balanço final estiver abaixo ou em igualdade com este valor, o projeto do sistema é visto como um sucesso.

A *facilidade de integração* de cada componente pode ser vista como a união entre dois aspetos: a forma como as suas entradas e saídas podem ser conetadas entre posteriores e subsequentes blocos, ou seja, se esta conetividade é garantida através de elementos de *hardware* ou *software* que estejam disponíveis prontamente, e a simplicidade de manuseamento, quer seja em termos físicos,

Figura 3.2: Antena YHA-66

como, por exemplo, condições favoráveis para soldar o componente a olho nu ou a montagem do componente seja acessível, quer seja em termos de programação ou instalação de *software*.

O *custo* do sistema será avaliado em termos monetários, tendo em conta que já é monitorizado o custo de potência. O custo total do sistema deverá rondar entre os €100 e os €150 para que a tentativa de projeto seja bem sucedida.

A *disponibilidade* de cada componente do sistema escolhido também tem o seu peso. Esta disponibilidade será avaliada tendo como referência três situações possíveis, ordenadas de melhor para pior caso: acesso ao componente dentro das instalações da Faculdade de Engenharia, acesso ao componente através da aquisição do mesmo a partir dos distribuidores com os quais a Faculdade de Engenharia colabora normalmente e o respetivo tempo esperado de chegada, e a total inacessibilidade ao componente.

O estabelecimento deste conjunto de critérios permite fazer uma análise qualitativa de cada dispositivo ou *software* eleito por forma a justificar cada decisão.

3.2.1 Antenas

São necessárias quatro antenas para efetuar a aquisição de sinal necessária para o desenvolvimento deste projeto. As antenas adequadas devem ter um padrão de radiação omnidireccional e funcionar na banda estabelecida nos requisitos do sistema. Para desempenhar esta função foram escolhidas quatro antenas YHA-66 (ver figura 3.2) do fabricante *Yaesu*, antenas essas que fazem parte do equipamento VX-3R [14] do mesmo fabricante.

Este equipamento não satisfaz completamente o requisito do sistema referente à banda de frequências. A antena YHA-66 funciona na banda dos 144 MHz e na banda dos 430 MHz. Apesar de se revelar uma limitação ao sistema, é possível utilizar este equipamento, uma vez que se encontra prontamente disponível e as suas bandas de operação encontram-se dentro da banda definida.

A conexão à antena YHA-66 é feita a partir da sua entrada SMA. Uma vez que este tipo de conexão é bastante comum na área da radiofrequência e dada a existência de conetores e cabos deste tipo na Faculdade de Engenharia, a integração deste dispositivo no sistema é facilmente assegurada.

O custo deste equipamento é nulo devido a estar disponível para o uso no âmbito de projetos da Faculdade de Engenharia.

A disponibilidade destas antenas na Faculdade de Engenharia faz com que o critério esteja cumprido, visto que estes equipamentos podem ser utilizados dentro das imediações da Faculdade de Engenharia a qualquer altura.

Apesar de não respeitar todos os critérios estipulados, este equipamento será integrado no sistema.

3.2.2 Comutador de antenas

O dispositivo escolhido para fazer a seleção das antenas é um *chip* da empresa *Analog Devices* com a designação de multiplexador 4:1 de banda larga, ilustrado na figura 3.3. Segundo a *datasheet* disponibilizada pelo fabricante [15], a largura de banda do comutador é de cerca de 2.5 GHz, o isolamento é de cerca de -37 dB a um frequência de 1 GHz, as perdas por inserção são de cerca de 1.1 dB a 1 GHz e as entradas para as antenas estão adaptadas a 50 Ω. Estas características garantem um bom desempenho para aplicações de radiofrequência até 1 GHz, o que é suficiente para a banda de frequências requerida para o sistema em projeto. O facto de haver um documento [15] com as especificações do produto, as aplicações típicas e informação sobre como controlar este *switch* torna o componente adequado aos requisitos do sistema.

A utilização da tecnologia CMOS por parte deste dispositivo permite garantir o seu funcionamento com um baixo consumo de potência. A alimentação situa-se entre os 1.65 V e os 2.75 V com um consumo máximo de corrente de 1 µA. Este consumo de potência adequa-se ao critério de baixo consumo estabelecido.

O circuito integrado está disponível em dois tipos de pacotes: pacote TSSOP de 20 pinos e pacote LFCSP de 20 pinos. A alternativa que torna mais fácil a integração no sistema é o pacote TSSOP de 20 pinos. Apesar dos pinos serem de dimensões milimétricas (0.3 mm de largura de cada pino), é possível soldá-los a olho nu ou com ajuda de um microscópio. Já o controlo deste componente é assegurado pela implementação de uma tabela de verdade disponibilizada na *datasheet* do fabricante. Esta tabela pressupõe três entradas digitais: $\overline{EN}$, A_0 e A_1. Mais informação sobre estas entradas digitais será fornecida aquando da implementação do sistema. A importância desta tabela de verdade é que é facilmente controlada através de um microcontrolador. O pacote TSSOP de 20 pinos disponível e a simplicidade de implementação do controlo do *switch* contribuem a favor do componente em termos de facilidade de integração.

O preço de cada unidade na data de aquisição era de €4.19 no distribuidor *Mouser*. Consequentemente, os critérios de disponibilidade e custo são satisfeitos.

3.2.3 Microcontrolador

A figura 3.4 ilustra a placa de desenvolvimento MSP-EXP430G2 *Launchpad* baseada no microcontrolador MSP430, fabricada pela empresa *Texas Instruments*. Este componente foi selecionado para arcar com a função de atuar como o microcontrolador responsável pelo controlo do

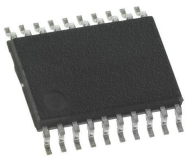

Figura 3.3: *Chip* comutador de antenas ADG904 baseado em tecnologia CMOS

switch de antenas, assim como para fornecer alimentação a esse mesmo dispositivo.

O fabricante, a empresa *Texas Instruments*, disponibiliza o documento técnico com as especificações desta placa de desenvolvimento [16]. O documento técnico é completo, apresenta uma descrição detalhada do *hardware* presente na placa e contêm instruções sucintas acerca da utilização do *software* disponível para esta unidade. Estes aspetos do componente estão em concordância com os requisitos do sistema previamente estabelecidos.

A alimentação deste produto é feita através da aplicação de 5 V numa entrada *mini USB* ou de aplicação direta nos pinos disponíveis. O consumo previsto para o microcontrolador com um CPU RISC de 16 bit e registos também de 16 bit presente na placa, o MSP430G2553, é de 230 µA a funcionar a 1 MHz [17] com a uma tensão aplicada de 2.2 V. O consumo de potência não afeta em demasia do orçamento estipulado.

O *Launchpad* oferece 20 pinos para entradas ou saídas. Estes pinos podem ser utilizados fisicamente através de *jumper wires* com entrada fêmea. Estes fios de cobre estão disponíveis nos laboratórios da Faculdade de Engenharia. A *interface USB* não só permite o fornecimento de potência como também serve o propósito de *interface* de programação. O ambiente de desenvolvimento *Code Composer Studio* [18] é um *software* disponibilizado de forma gratuita pela *Texas Instruments* dedicado à programação de componentes da empresa, incluindo os microcontroladores da série MSP430. Este IDE integra bibliotecas de programação que permitem o desenvolvimento de programas em linguagem C de forma fácil. Também o processo de escrita dos programas desenvolvidos na memória *flash* de 16 kilobytes dos microcontroladores é assegurado pelo *Code Composer Studio*. Existe uma alternativa a este *software* desenvolvida por uma comunidade *online* chamada *Energia* [19]. O ambiente de desenvolvimento *Energia* é um programa *open source* que permite a programação de microcontroladores MSP430 e a consequente escrita na memória dos mesmos. Estas duas alternativas de *software* de desenvolvimento para os microcontroladores MSP430 tornam a programação deste tipo de dispositivos bastante acessível. Como tal, está garantida a facilidade de integração do dispositivo no sistema.

A placa de desenvolvimento MSP-EXP430G2 *Launchpad* foi adquirida de forma rápida (tempo de espera de 2 semanas) a um preço de €9.39 ao distribuidor *Farnell*. Logo, os critérios de custo e disponibilidade são cumpridos.

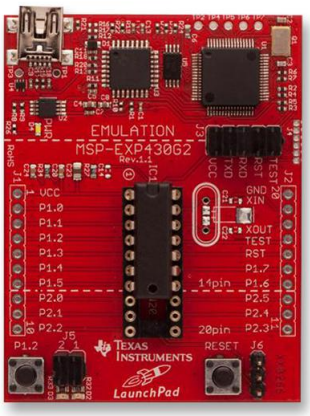

Figura 3.4: Placa de desenvolvimento TI MSP-EXP430G2 *Launchpad* baseada no microcontrolador MSP430

3.2.4 *Dongle RTL*

O *dongle USB* adquirido está ilustrado na imagem 2.5. Este componente é uma *pen* concebida para receber canais de televisão digital terrestre (DVB-T), canais de rádio digital (DAB) e emissões de rádio analógico (FM). No entanto, tal como é explicado na subsecção 2.2.2.1, este tipo de dispositivos baseados no *chip* RTL2832u [13] pode ser usado como um *software defined-radio*.

O sintonizador incorporado no *dongle* é o modelo R820T do fabricante *Rafael Microelectronics Inc*. Segundo a *datasheet* deste circuito integrado [20], este sintonizador funciona para uma banda de frequências dos 42 MHz até aos 1002 MHz. Apesar da gama de frequências ser mais estreita do que o estabelecido pelos requisitos do sistema, é possível utilizar este dispositivo. A *datasheet* do sintonizador está disponível *online* e é informativa. No entanto, a documentação técnica do desmodulador RTL2832u fabricado pela empresa *Realtek Semiconductor Corp.* não é disponibilizada. Tendo em conta que ambos os módulos, o sintonizador e o desmodulador, são os pontos centrais deste *dongle*, e apenas um deles oferece informação técnica, os requisitos do sistema em termos de documentação e acessibilidade não estão completamente satisfeitos. A interação com este dispositivo não é a ideal.

Segundo a *datasheet* do sintonizador [20], este *tuner* tem um consumo de potência de cerca de 178 mA alimentado com uma tensão de 3.3 V. A ausência de documentação técnica acerca do desmodulador faz com que o seu consumo de potência seja incerto. No entanto, Antti Palosaari, o *developer* que descobriu a possibilidade de se usar este tipo de *dongles* como SDR, fez algumas medições [21] e determinou a corrente total de 313 mA a uma tensão nominal de 5 V para um *dongle* com o sintonizador R820T e o desmodulador RTL2832u. Usar-se-á este valor como referência para fazer o balanço total de potência. Este valor de potência é algo abusivo para aquilo que foi projetado, mas pensa-se que será facilmente suportado.

O componente apresenta uma ligação USB 2.0, compatível com a maioria dos equipamentos hoje em dia. A entrada RF é um conetor SMA (SubMiniature version A) fêmea logo, a ligação a uma antena pode ser efetuada facilmente, visto ser um conetor bastante corrente em aplicações de

radiofrequência. Existem cabos SMA e várias derivações desse mesmo conetor disponíveis nos laboratórios da Faculdade de Engenharia. A conexão USB 2.0 possibilita o *dumping* de amostras em formato I/Q no dispositivo com o qual a *pen* se encontra emparelhada. A facilidade de integração está então garantida.

O preço de cada unidade está dentro da gama €10 - €20. A unidade está disponível em várias plataformas de venda *online*, como *Amazon* ou *Ebay*. A data de espera é de cerca de 2 semanas. A conformidade com os critérios de preço e disponibilidade tornam este dispositivo apto para o integração no sistema.

3.2.5 Aquisição de dados

A aquisição dos dados é delegada para o equipamento *Raspberry Pi Model B+*, presente na figura 3.5. Este dispositivo é bastante popular nos dias que correm, maioritariamente devido ao facto de ser um computador barato com um CPU ARM de 700 MHz integrado num processador *Broadcom* BCM2835 [22] e um extenso suporte para periféricos, perfeito, como tal, para prototipagem e projetos amadores. Existe uma enorme comunidade que contribui para o desenvolvimento de *software* para esta plataforma, incluindo o desenvolvimento de sistemas operativos livres baseados em *Debian* ou *Arch Linux*, como é o caso do sistema operativo *Raspbian* [23]. Aliando ao *Raspberry Pi Model B+* o *software rtl-sdr* criado por uma comunidade *online* e baseado na biblioteca *librtlsdr* (código fonte aberto no repositório [24]) [25], é possível adquirir e guardar em memória os dados fornecidos pelo *dongle USB*.

A existência de uma extensa comunidade *online* [26] dedicada ao *Raspberry Pi* e o fácil acesso à documentação [27] fazem com que este componente esteja dentro dos parâmetros estabelecidos para o sistema. O pacote de *rtl-sdr* também se revela uma ótima escolha para o sistema visto que o seu código fonte é livre [28].

O consumo de potência do *Raspberry Pi Model B+* tem como teto máximo uma corrente de 1 A a uma tensão de 5 V aplicada na sua entrada *micro USB* [29]. Este parece ser o elemento do sistema cujo consumo de potência é maior. No entanto, acredita-se que é sustentável para a carga prevista.

O *Raspberry Pi Model B+* dispõe de 4 entradas USB 2.0, é alimentado através de uma entrada *micro USB* e possui 2 entradas *Ethernet* 10/100 RJ45 [22]. Estas características tornam possível o emparelhamento com o *dongle RTL* e o acesso cablado à Internet. A instalação do pacote de *software rtl-sdr* é feita seguindo um conjunto de simples instruções [30]. Estão garantidas as regras estabelecidas para a facilidade de integração.

O *software rtl-sdr* é livre e está disponível em [25]. O *Raspberry Pi Model B+* foi adquirido por um preço de €32.78 e demorou cerca de duas semanas a ser entregue. Os critérios de preço e disponibilidade estão satisfeitos, tendo em conta o reduzido preço do *Raspberry Pi Model B+*.

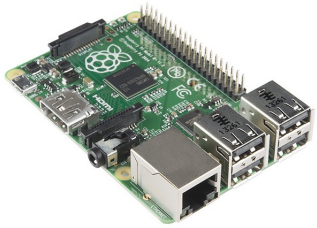

Figura 3.5: *Raspberry Pi Model B+*

3.2.6 Processamento e visualização de dados

O processamento e a visualização dos dados estão ao encargo do *software* de computação técnica e ambiente de desenvolvimento *Matlab* [31]. Este ambiente de desenvolvimento é uma ferramenta poderosíssima em termos de processamento de dados, com um pacote especialmente dedicado ao processamento digital de sinal [32], algo que vai ao encontro das necessidades deste projeto, assim como em termos de visualização de dados [31].

A utilização do ambiente de desenvolvimento *Matlab* vai ao encontro dos requisitos do sistema devido à quantidade de documentação disponibilizada pela *Mathworks* [33], empresa responsável pela conceção deste produto.

A facilidade de integração deste *software* no sistema é alcançada devido à capacidade de integração com programas escritos em linguagens como C, C++, Java e .NET, e pela possibilidade de ser utilizado para conceber uma *interface* gráfica de interação com utilizadores através do *software GUIDE* incorporado no ambiente de desenvolvimento [31]. Isto torna o *software* perfeito para a visualização de dados.

O preço de uma licença de *Matlab* não constitui um problema, uma vez que a Faculdade de Engenharia disponibiliza licenças deste ambiente de desenvolvimento para propósitos académicos aos seus alunos. Logo, trabalhando dentro da Faculdade de Engenharia ou ligando um computador à sua rede interna, o *software* está prontamente disponível.

É relevante mencionar a utilização do *software gqrx* [34] (ver 3.7) e *Airspy SDR#* [35] (ver 3.6) para visualização e gravação de dados para avaliar o sucesso das experiências efetuadas ao longo da implementação do projeto. Cada um destes programas oferece uma interface gráfica de utilizador com um analisador de espetros e um sintonizador e permite interagir com o *dongle RTL* para alterar os parâmetros de receção.

3.2.7 Balanço de potência

Uma vez que já se encontram estabelecidos os dispositivos responsáveis por cada um dos blocos do diagrama 3.1, é tempo agora de elaborar o balanço total de potência.

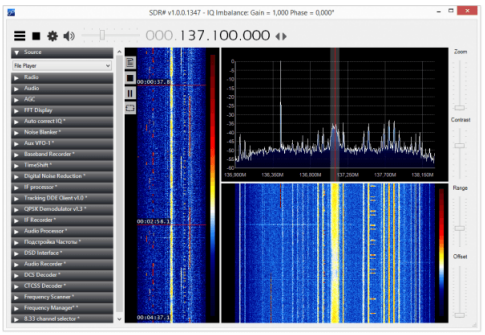

Figura 3.6: *Airspy SDRSharp*

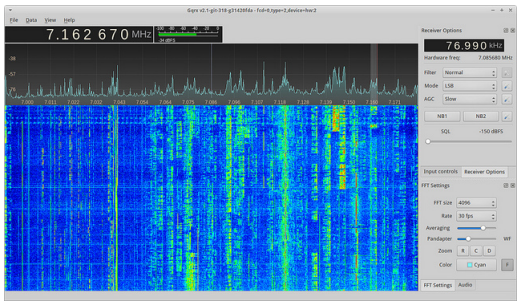

Figura 3.7: *Gqrx*

Assumindo o pior cenário, ou seja, consumo máximo de corrente por parte de todos os componentes do sistema, elaborou-se a tabela 3.1.

A partir da tabela 3.1, é possível inferir que o consumo de corrente do sistema fica abaixo do valor inicialmente estipulado de 3 A, somando um total de 1.3132 A.

3.2.8 Custo total do sistema

Um dos parâmetros cruciais para o sistema é o custo total do sistema. A tabela 3.2 apresenta o somatório do preço de cada componente do sistema.

Torna-se claro que não se atinge o orçamento entre os €100 e os €150 ao qual o custo total do sistema estaria comprometido. Este facto é visto como um sinal positivo para o projeto do sistema.

3.3 Processamento de Sinal

Tal como estabelecido previamente no presente documento, o objetivo deste projeto é localizar emissores rádio através de *software-defined radio*. Depois de garantidas as condições de aquisição de dados, o processamento de amostras digitais provenientes do *dongle RTL* evidencia-se como o passo fundamental para inferir conclusões sobre a localização de um emissor de interesse. A técnica de localização de emissores rádio implementada é baseada em TDOA.

O diagrama 3.8 revela a sequência de passos utilizada para inferir a localização do emissor a partir de dados capturados pelo *dongle RTL*.

Cada um dos blocos é explicado em seguida:

Tabela 3.1: Balanço de potência do sistema

Componente	**Corrente consumida**
ADG904	1 µA
MSP430G2553	230 µA
Dongle RTL	313 mA
Raspberry Pi Model B+	1 A
Total	1.3132 A

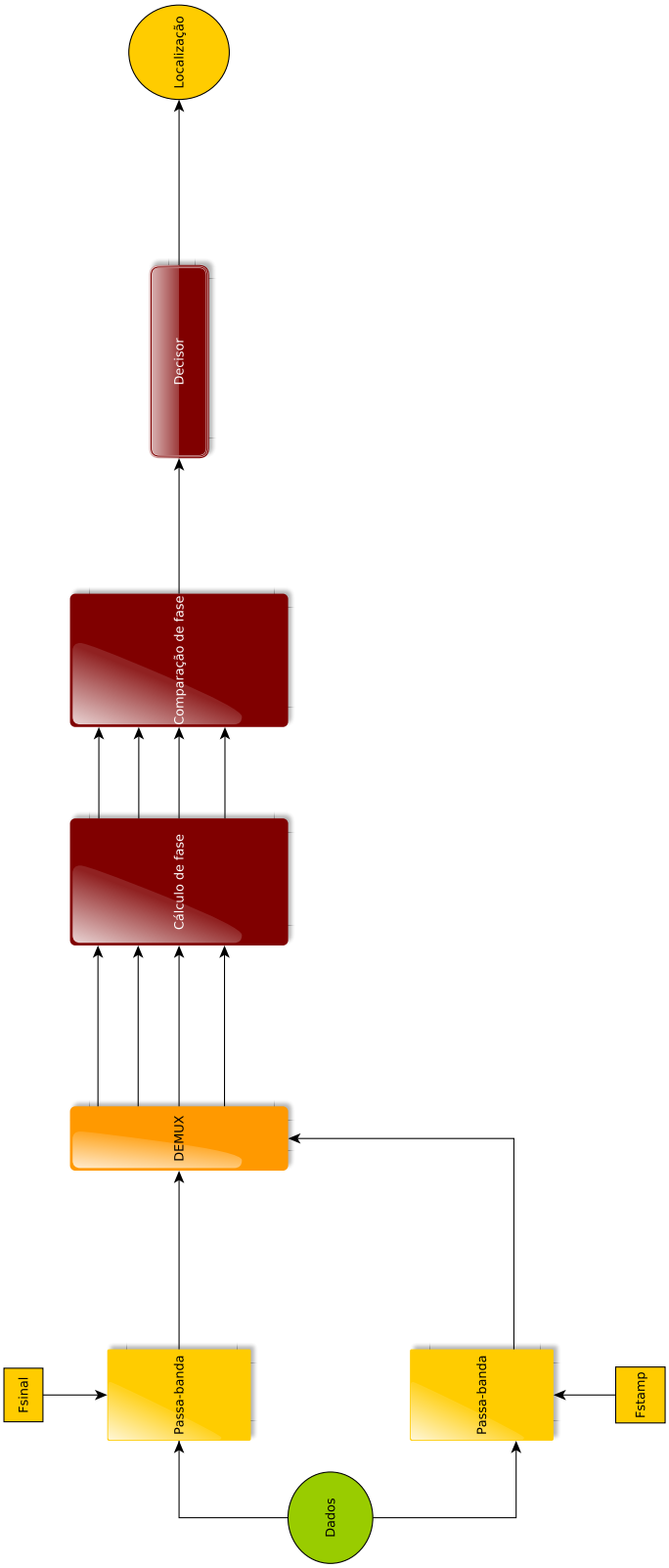

Figura 3.8: Processamento efetuado aos dados capturados

Tabela 3.2: Custo total do sistema

Componente	Preço
ADG904	€4.19
MSP430G2553	€9.39
Dongle RTL	€20
Raspberry Pi Model B+	€32.78
Total	€66.36

Dados Bloco representativo dos dados capturados pelo *dongle RTL* em formato I/Q, isto é, quadratura e fase.

Filtro Passa-banda Existem dois filtros passa-banda no diagrama 3.8. O filtro passa-banda que recebe o parâmetro F_{sinal} é responsável por deixar passar as componentes da frequência de interesse presentes nos dados. O outro filtro passa-banda, aquele que recebe o parâmetro F_{stamp}, é responsável por filtrar um *timestamp* presente no sinal que indica os instantes temporais no qual a antena 1 está ligada. Este é o mecanismo de sincronização adotado para o sistema.

DEMUX O bloco DEMUX representa a desmultiplexagem feita aos dados. Aqui serão separados os dados das quatro antenas do sistema. Essa separação é feita através do conhecimento dos instantes em que a antena 1 está a captar.

Cálculo de fase Este bloco computa a fase dos dados de cada uma das antenas.

Comparação de fase Aqui são comparadas as fases de cada uma das antenas.

Decisor O bloco decisor escolhe qual é a direção do emissor.

Localização Dados relativos à localização do emissor.

Cada um dos blocos mencionados pressupõe uma análise cuidada por forma a tornar a implementação mais imediata e fácil. Essa análise será feita nas subsequentes subsecções.

3.3.1 Dados

O *chip* RTL2832u contêm um ADC que realiza amostragens a uma frequência máxima de 3.2 milhões de amostras por segundo (podendo não ser estável a esta frequência de amostragem) e cuja resolução efetiva de bits é de 7 bits [36] e que coloca à sua saída dados de I e Q em formato de inteiro de 8 bits sem sinal (*uint8* em linguagem C ou Matlab). Este formato de saída dos dados é obtido usando o módulo de *software rtlsdr* [28].

Ao longo deste trabalho, ocorreram capturas de dados obtidas através de 3 tipos de *software*: o módulo *rtlsdr*, o *Airspy SDR#* e o *gqrx*. Tal como foi referido anteriormente, o módulo *rtlsdr* retorna amostras I e Q em formato de inteiro de 8 bits sem sinal. No entanto, ambos os outros programas de visualização e captura de dados permitem a captura de sinal em formato *WAVE* [37] que é importado para o Matlab como tipo de dados de vírgula flutuante de dupla precisão de 64 bits com sinal (*double* [38]).

O processamento sobre estes dados pressupõe um conjunto de amostras I e Q, ou seja, com valores reais e imaginários. Por isso, a estrutura de dados usada no *software Matlab* é *complex double*, uma vez que as amostras são processadas de forma a que cada uma delas represente um número complexo. Esta representação das amostras revela-se prática, uma vez que serão aplicadas transformadas de Fourier e outras operações que beneficiam das propriedades desta transformada relativamente a números com parte real e parte imaginária.

3.3.2 Filtros

Os filtros implementados no decorrer deste projeto são filtros digitais de resposta impulsional finita, conhecidos na literatura como filtros *Finite Impulse Response* ou filtros FIR. O filtros FIR permitem atingir uma resposta em frequência com fase linear, algo que pode ser reconhecido como um atraso constante, simplificando o projeto destes filtros no que toca à aproximação de uma resposta em magnitude desejada [39]. Este facto faz com que não seja necessário lidar com o problema da distorção de fase, havendo apenas o foco na resposta em magnitude do filtro desejado. Como se tenciona fazer uma análise ao conteúdo de fase das amostras capturadas, o projeto de filtros FIR parece o mais adequado neste contexto prático.

Quando as amostras relevantes são extraídas do *dongle RTL* e fornecidas à plataforma de visualização e processamento de dados, é necessário fazer a filtragem do sinal de interesse e do sinal de *timestamping* usado para sincronização de canais, removendo as outras componentes de frequência presentes nos dados. Para tal é concebido um filtro FIR passa-banda de ordem 1024 com as frequências de corte especificadas de forma a abranger a largura de banda do sinal de interesse e do sinal de *timestamping*. A resposta em magnitude e fase do filtro a aplicar está representada em 3.9 onde as frequências estão normalizadas relativamente à frequência de Nyquist, ou seja, $f_{Nyquist} = f_{amostragem}/2 = 1.0$. No que toca à magnitude do filtro, a banda passante apresenta-se com uma magnitude de 0 dB enquanto que na banda de rejeição o filtro atenua -70 dB às frequências próximas da banda de passagem e atinge os -90 dB às frequências mais afastadas da banda de passagem.

A informação sobre as bandas de passagem relativas ao sinal capturado e ao sinal de sincronismo será definida de acordo com a aplicação em causa, daí a falta de estipulação desses mesmos valores nesta secção.

3.3.3 Desmultiplexagem

A desmultiplexagem, também denominada de DEMUX ao longo deste documento, representa a recuperação da informação temporal referente a cada canal, assumindo que cada uma das quatro antenas é um canal.

Em termos de processamento de sinal, a desmultiplexagem de canais é feita através do seguinte conjunto de procedimentos:

- Geração local de onda quadrada à frequência do sinal de *timestamping*

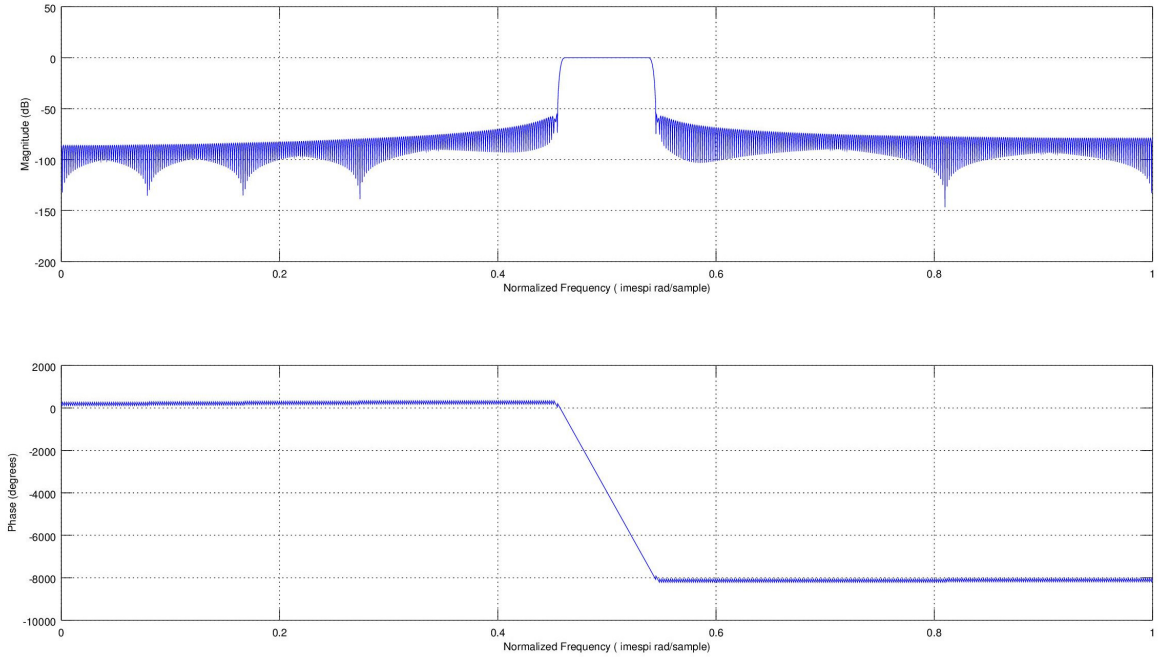

Figura 3.9: Magnitude e fase do filtro FIR passa-banda de ordem 1024

- Correlação entre os impulsos quadrados gerados localmente e os dados capturados
- Determinação do atraso (em amostras) do sinal gerado localmente e o sinal de *timestamping*
- Geração de quatro sinais selectores de forma de onda quadrada com os relativos atrasos entre eles
- Multiplicação dos seletores pelos dados capturados

A *geração local de onda quadrada à frequência de timestamping* é efetuada como uma réplica do sinal utilizado para injetar informação sobre qual canal se encontra a transmitir a cada instante. Este sinal é uma onda quadrada gerada pelo microcontrolador à frequência de um quarto da frequência de comutação do *switch* de antenas.

A *correlação entre os impulsos quadrados gerados localmente e os dados capturados* é a operação executada para analisar as semelhanças entre o sinal de *timestamping* presente nas amostras e a onda quadrada de referência. Esta operação é calculada através da multiplicação da *Fast Fourier Transform* (FFT) do sinal de sincronização com a onda quadrada gerada localmente.

A *determinação do atraso (em amostras) do sinal gerado localmente e o sinal de timestamping* pressupõe a procura da amostra onde a operação de correlação tem a maior magnitude. Este valor representa o atraso da onda gerada localmente relativamente ao sinal de sincronização.

A *geração de quatro sinais selectores de forma de onda quadrada com os relativos atrasos entre eles* é o procedimento de realização dos seletores de canais. Este procedimento pressupõe a geração de quatro ondas quadradas com amplitudes 0 ou 1 com frequência de um quarto da frequência de comutação desfasados entre cada um de cerca de $\pi/2$ ou 90°, tendo em conta o atraso

computado anteriormente. Estes sinais funcionam como seletores de canais, uma vez aplicados ao sinal de interesse.

A *multiplicação dos seletores pelos dados capturados* é a operação efetuada para selecionar as amostras de cada uma das antenas, através do produto entre cada um dos seletores pelos dados capturados.

3.3.4 Cálculo de fase

O cálculo de fase é efetuado em cada um dos conjuntos de dados referentes a cada canal para que exista uma posterior comparação entre canais. Este cálculo é facilmente executado na plataforma *Matlab*, uma vez que os dados se apresentam como números complexos.

3.3.5 Comparação de fase

A comparação de fase entre os sinais capturados pelos diferentes canais é feita através do cálculo da diferença de fase entre cada um dos canais relativamente aos restantes. Logo, são efetuados 6 cálculos de diferença de fase: a diferença de fase entre a antena 1 e a antena 2, entre a antena 1 e a antena 3, entre a antena 1 e a antena 4, entre a antena 2 e a antena 3, entre a antena 2 e a antena 4 e entre a antena 3 e a antena 4. Esses valores são posteriormente processados pelo bloco decisor do presente sistema de processamento de sinal.

3.3.6 Decisor

O decisor tem a função de computar a direção de chegada do sinal incidente no agrupamento de antenas através de uma análise dos valores de diferença de fase entre cada uma das antenas. Até à altura de redação deste documento, não foi possível derivar um algoritmo para desempenhar as funções de decisor.

3.3.7 Localização

O bloco de localização presente no diagrama 3.8 é responsável pela visualização dos dados. A visualização dos dados pressupõe uma tradução dos dados vindos do bloco do decisor para um plano a duas dimensões definido como aquele representado na figura 3.10.

3.4 Conclusão

Ao longo deste capítulo, foi apresentada uma visão geral sobre o sistema em projeto, demonstraram-se e justificaram-se as escolhas de *hardware* e *software* e introduziu-se a abordagem de processamento de sinal que se julga ser a melhor opção para o sistema em desenvolvimento.

É de valor salientar que ao longo da implementação do sistema, serão incorporados módulos de *hardware* e *software* que estão para além do escopo principal deste projeto e, como tal, serão referidos ao longo da descrição do processo de desenvolvimento do projeto.

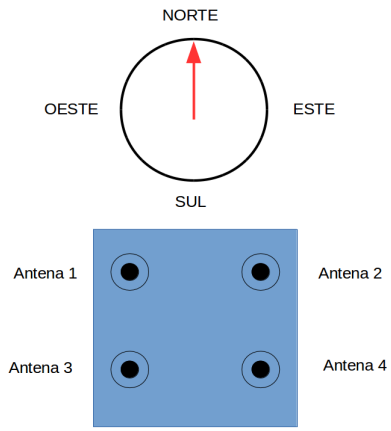

Figura 3.10: Esquema de referência para o plano 2D de visualização da direção de emissor

Capítulo 4

Implementação do Sistema

Este capítulo obedece ao propósito de expor o leitor à implementação prática do sistema. O trabalho apresentado ao longo deste documento incidiu maioritariamente na capacidade de rapidamente prototipar soluções e colocá-las em funcionamento quanto antes. Como tal, serão apresentados os esquemáticos e *layouts* desenvolvidos para o *hardware* deste projeto, assim como as arquiteturas utilizadas em termos de *software*.

4.1 Caso de uso principal

O caso de uso principal para o qual se pretende implementar este sistema é o acoplamento do sistema de aquisição de dados ao teto de um veículo automóvel. Para tal, é necessário que o sistema de aquisição seja alimentado pela bateria do automóvel através da ficha do isqueiro que, tipicamente, apresenta valores de tensão que rondam os 12 V. É também necessário garantir que o sistema tenha ligação à Internet. Este requisito é cumprido através da utilização do *modem D-LINK 4G USB adapter* DWM-221, que permite o acesso à rede 4G no caso da existência de um cartão *SIM* com subscrição a uma operadora de telecomunicações para o serviço 4G, emparelhado com um *router D-LINK* DWR-116 por forma a oferecer rede *Wifi* à plataforma de processamento e visualização de dados (idealmente, um computador portátil) e a oferecer acesso à rede ao sistema de aquisição através de um cabo *Ethernet*. Devido ao uso de acesso cablado à rede, é possível alimentar o sistema de aquisição através de uma ligação *Ethernet*, aplicando o mesmo princípio usando em *Power over Ethernet*.

O projeto não atingiu o ponto de implementação necessário para pôr em prática este caso de uso. No entanto, julga-se ser relevante que o leitor tenha este caso de uso como o contexto justificativo das decisões tomadas ao longo da implementação deste sistema.

4.2 PCB para Comutador de antenas

Para integrar o *switch* de antenas ADG904, referido em 3.2.2, no sistema, é necessário o desenvolvimento de uma *printed circuit board* para alojar o dispositivo e realizar as devidas ligações.

O programa escolhido para desenvolver a PCB de comutação de antenas foi o *software open-source* de *EDA KiCad* [40].

4.2.1 Esquemático

Na primeira fase de desenvolvimento desta placa de circuito impresso, elaborou-se um esquemático de todos os componentes que incorporam este dispositivo.

No esquemático estão presentes os seguintes elementos:

3V3 Pino responsável pela alimentação do circuito a um potencial de 3.3 V.

GND Pino de *ground*.

EN Pino de controlo do *enabler* do comutador ADG904.

A0 Pino de controlo da entrada digital A0 do *switch* ADG904.

A1 Pino de controlo da entrada digital A1 do *switch* ADG904.

MCP1703 LDO MCP1703 da *Microchip*. Descrição em 4.2.1.1

ADG904 *Switch* de antenas da *Analog Devices*. Descrito em 3.2.2.

SMAE1-5 Conetores RP-SMA para ligação com as antenas de captura e saída do comutador. Dispositivo físico representado em 4.2.

A elaboração de um esquemático é de extrema importância para o *layout* da placa de circuito impresso devido à capacidade de se gerar uma *netlist*. Essa *netlist* garante que os componentes, aquando da passagem para o ambiente de desenvolvimento de *layout* em formato de *footprint*, obedecem às ligações estabelecidas previamente no esquemático. Os *footprints* de componentes dizem respeito a um modelo físico do *software* de desenvolvimento, onde os pinos e os pacotes desses dispositivos se encontram com as dimensões exatas. O *software* KiCad já inclui *footprints* de uso frequente na sua biblioteca, no entanto, foi necessário desenvolver um *footprint* para o comutador de antenas ADG904. Consultando a *datasheet* [15], produziu-se um *footprint* fiel às suas dimensões físicas.

4.2.1.1 LDO

As tensões de saída dos pinos de *GPIO* do microcontrolador MSP430 [16] e do *Raspberry Pi Model B+* [41] são de, respetivamente, 3.3 V e de 1.8 V até 3.6 V, dependendo do VCC utilizado para alimentar o MSP430. No entanto, o *switch* ADG904 requer uma tensão típica de utilização de 2.5 V [15]. Como tal, é necessário empregar algum tipo de regulação de linha antes do fornecimento de potência ao comutador.

Para exercer a função de regulação de linha necessária para o correto funcionamento do comutador de antenas ADG904, foi escolhido o LDO MCP1703 do fabricante *Microchip*. Segundo a

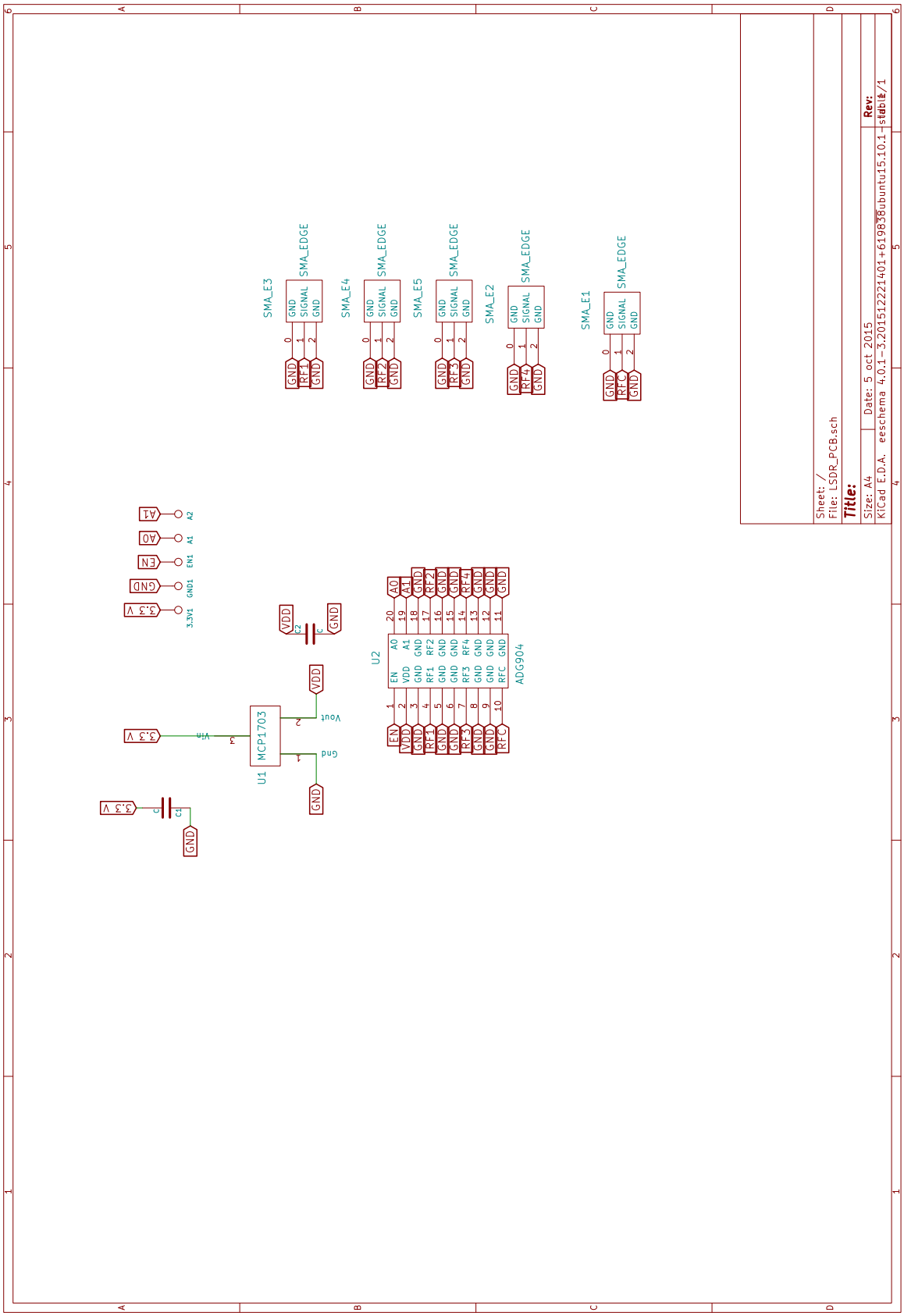

Figura 4.1: Esquemático da placa de circuito impresso de comutação de antenas

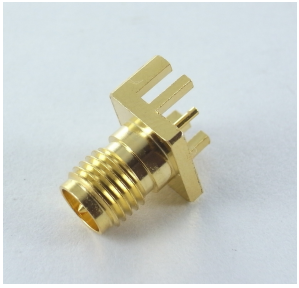

Figura 4.2: Conetor RP-SMA fêmea para PCB

datasheet fornecida pelo fabricante [42], este componente regula tensões de entrada desde os 2.7 V até aos 16 V, fornecendo uma corrente máxima de 250 mA a uma tensão de saída de 2.5 V. O dispositivo garante o correto funcionamento do comutador, uma vez que a corrente de polarização necessária é de 1 µA.

4.2.2 Tabela de verdade do comutador de antenas ADG904

A *datasheet* [15] do *switch* ADG904 providencia uma tabela de verdade com as informações relevantes para a implementação da comutação. Esta tabela de verdade está ilustrada na figura 4.3. O microcontrolador responsável pelo controlo deste dispositivo incorporará um programa que obedece a esta tabela de verdade. Portanto, o microcontrolador terá de fornecer 3 pinos ao comutador ADG904: o pino de *enabler*, o pino $A0$ e o pino $A1$.

4.2.3 *Layout*

O *layout* concebido para esta placa de circuito impresso encontra-se representado na imagem 4.4. As principais preocupações com este *design* foram: a largura das pistas que vão desde o comutador ADG904 até às saídas RP-SMA, a presença de um plano de massa ao comprimento e largura de cada pista por forma a garantir um caminho curto de fuga para correntes parasitas e o isolamento da parte analógica da parte digital. A largura de cada um das pistas onde circulam os sinais RF foi calculada por forma a ter uma impedância de 50 Ω a uma frequência de 1 GHz.

A1	A0	$\overline{EN}$	ON Switch[1]
X	X	1	None
0	0	0	RF1
0	1	0	RF2
1	0	0	RF3
1	1	0	RF4

Figura 4.3: Tabela de verdade do comutador de antenas ADG904 da *Analog Devices*

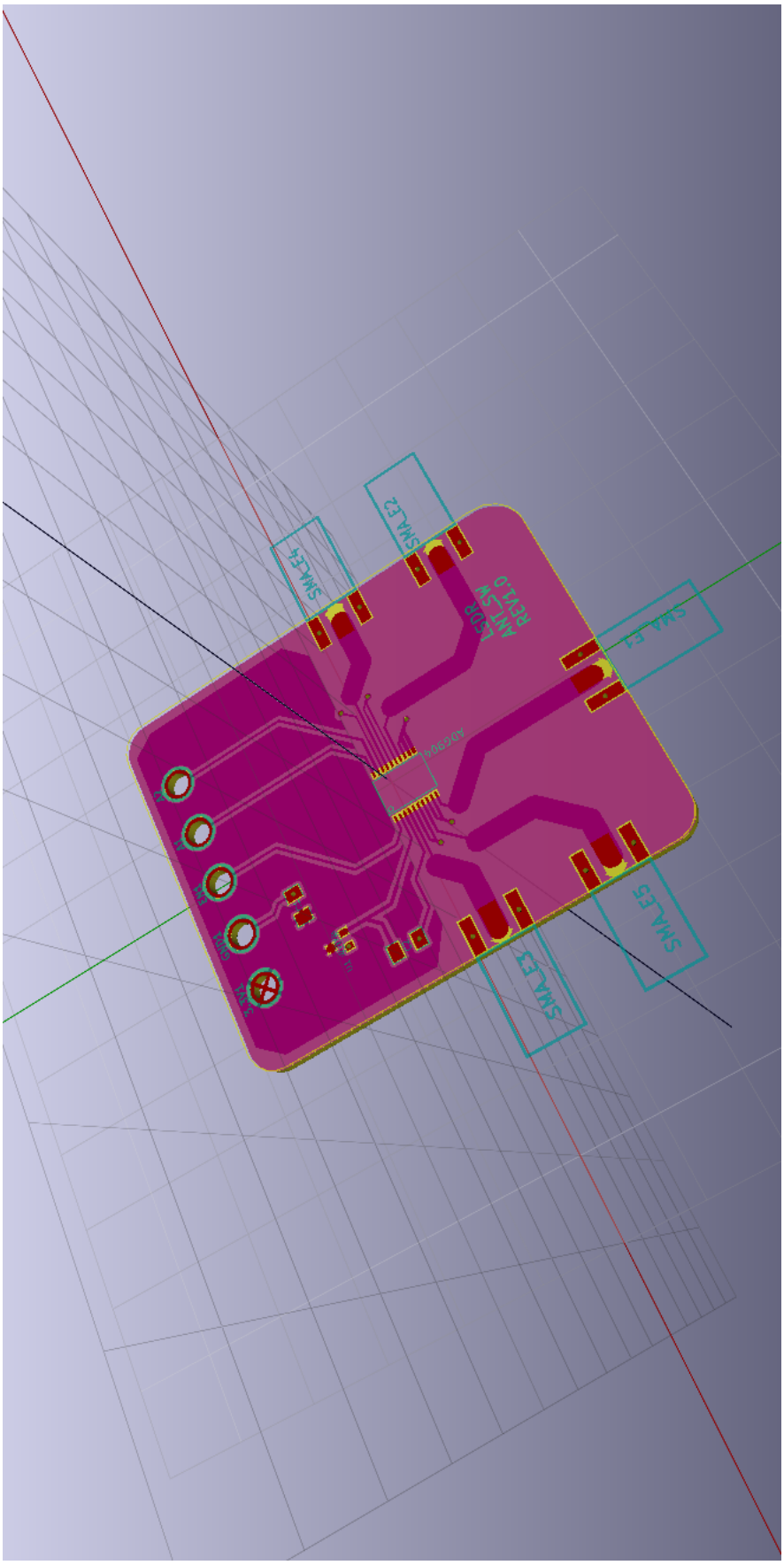

Figura 4.4: *Layout* da placa de circuito impresso de comutação de antenas

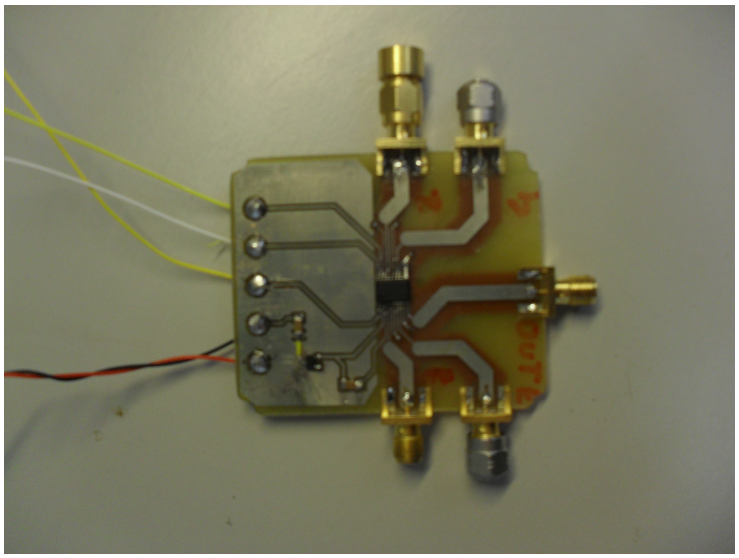

Figura 4.5: Placa de circuito impresso de comutação de antenas - Frente

4.2.4 Produto final

A placa de circuito impresso realizada encontra-se na figura 4.5, com o pormenor da parte posterior presente na figura 4.6.

4.2.4.1 Soldadura

O processo de soldadura dos componentes constituintes da placa de circuito impresso envolveu a utilização de uma lupa para alcançar a precisão necessária para soldar os componentes de menor dimensão.

4.3 Fornecimento de potência

Um dos aspetos cruciais do sistema é o fornecimento de potência ao sistema, tal como referido anteriormente 3.1. Para executar essa função, decidiu-se utilizar transferência de potência através de um cabo *Ethernet* por forma a alimentar o *Raspberry Pi Model B+* para que este consiga alimentar os módulos subsequentes. Como referência, usou-se o *standard IEEE 802.3af* [43], também conhecido como *Power-over-Ethernet* (PoE).

4.3.1 *Power-over-Ethernet*

O *standard IEEE 802.3af* [43], ou *Power-over-Ethernet*, define a utilização de cabos Ethernet para o transporte de dados e potência entre os dispositivos conetados. Funciona de forma análoga ao *phantom power* (48 V) de uma mesa de mistura que fornece potência para alimentar microfones de componentes ativos. O *standard* serviu apenas como referência, uma vez que as tensões utilizadas não vão ao encontro do estabelecido por esta mesma norma. O que foi usado foi a referência de pinos, ou cabos, dos cabos Ethernet usados para conetar os dispositivos. Essa

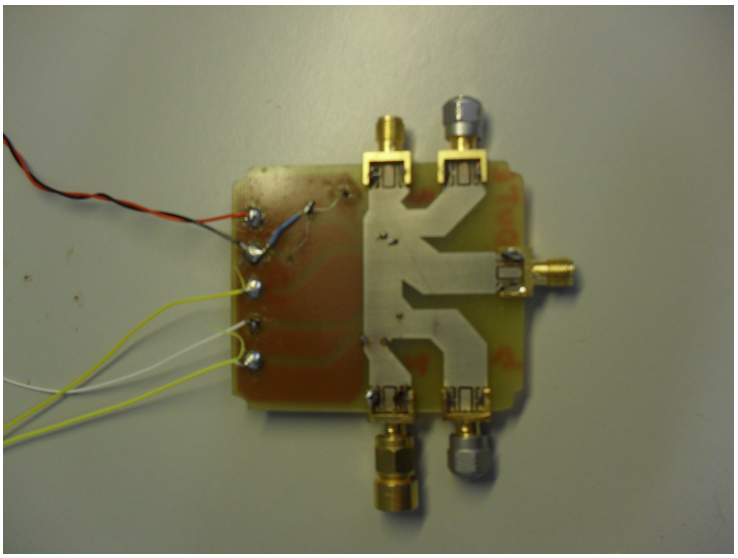

Figura 4.6: Placa de circuito impresso de comutação de antenas - Trás

informação foi obtida facilmente [44]. A tensão fornecida ao cabo de Ethernet é de 12 V, uma vez que o comprimento da ligação não ultrapassa os 3 metros.

4.3.2 *Traco Power* TSR 3-24150

O fornecimento de potência através de um cabo Ethernet transportando uma tensão de 12 V faz com que seja necessária uma conversão *stepdown* dc-dc de 12 V para 5 V, uma vez que essa é a tensão necessária para alimentar o *Raspberry Pi Model B+*. O abaixamento de tensão e a regulação de linha são efetuados pelo regulador comutado TSR 3-24150 do fabricante *Traco Power*, cuja gama de tensões de entrada vai desde os 10 V para 30 V com uma saída em tensão com valores entre os 5 V para 15 V com uma corrente máxima de 3 A [45]. Com este dispositivo, garante-se o fornecimento dos 5 V à entrada do *Raspberry Pi Model B+*. Ver figura 4.7.

4.3.3 Módulos físicos

Os módulos para o fornecimento de potência foram realizados em *veroboard*. Um dos módulos de PoE está representado nas figuras 4.8 e 4.9. Outro dos módulos está ilustrado pelas fotografias 4.10 e 4.11. Ilustra-se também a ligação do módulo PoE de fornecimento ao *Raspberry Pi Model B+* com o *Traco Power* TSR 3-24150 na figura 4.7.

4.4 Programação do Microcontrolador MSP430

O MSP-EXP430G2 *Launchpad*, introduzido na secção 3.2.3, inclui o microcontrolador MSP430 que faz o controlo do *switch* de antenas através da execução de um programa que implementa a tabela de verdade apresentada em 4.2.2.

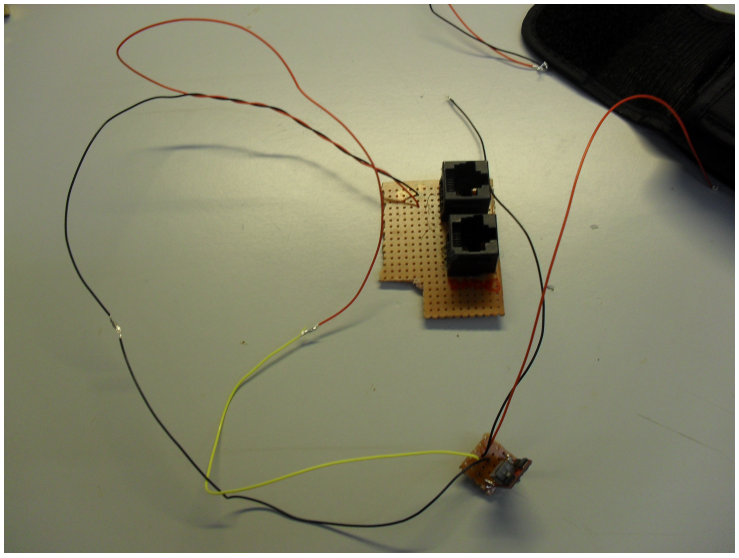

Figura 4.7: Módulo 2 de PoE + TRACO POWER

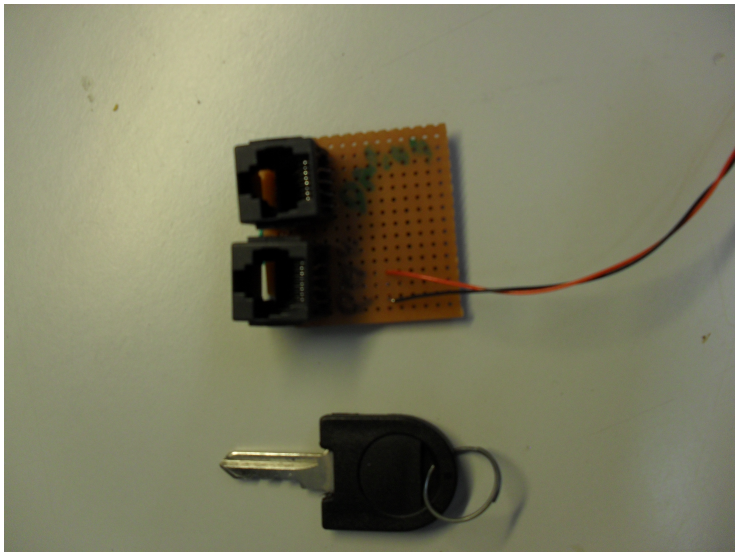

Figura 4.8: Módulo 1 de PoE - frente

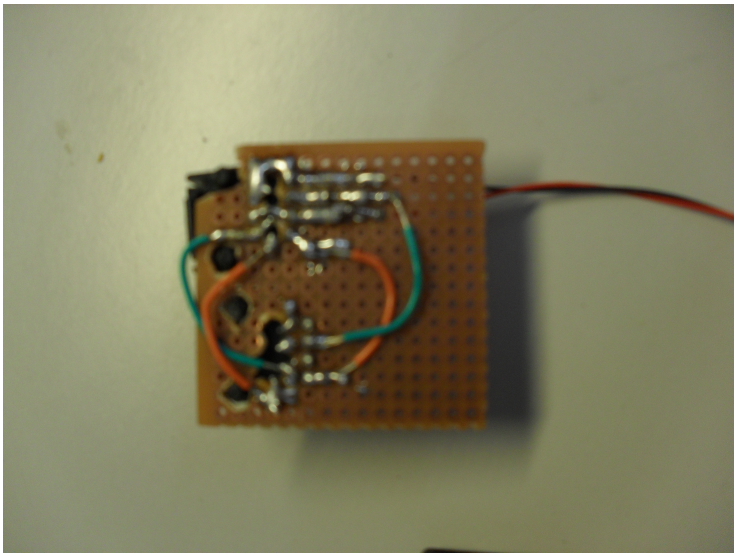

Figura 4.9: Módulo 1 de PoE - traseira

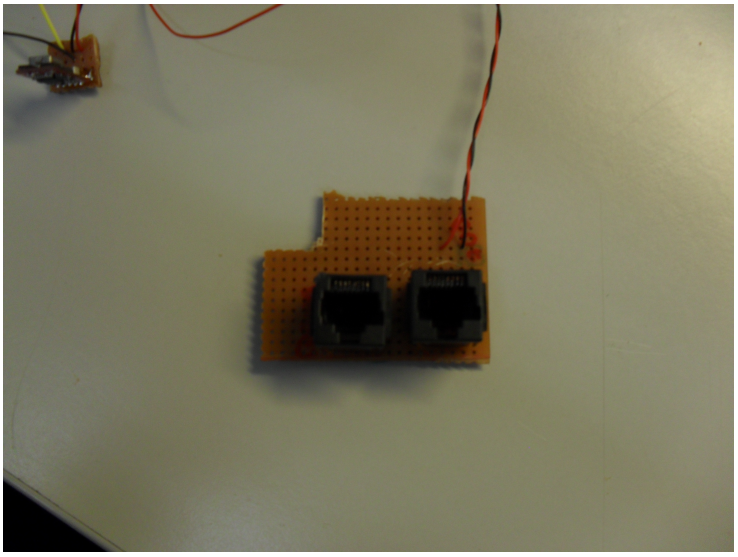

Figura 4.10: Módulo 2 de PoE - frente

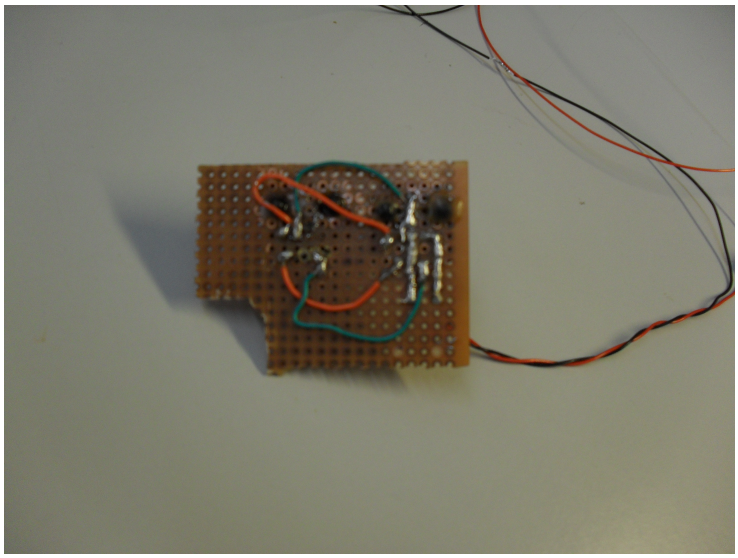

Figura 4.11: Módulo 2 de PoE - traseira

O primeiro passo para desenvolver um programa para o MSP430 é a instalação da plataforma de desenvolvimento recomendada pelo fabricante, o *software Code Composer Studio*. Em seguida, é necessário fazer a implementação do código de acordo com a sintaxe estipulada pelas bibliotecas presentes no IDE.

4.4.1 Implementação do código

A tabela de verdade apresentada em 4.2.2 pode ser implementada no microcontrolador MSP430 através da designação de pinos para representarem os valores de EN, $A0$ e $A1$. A utilização da biblioteca *msp430g2553.h* presente na plataforma de desenvolvimento *Code Composer Studio* permite a designação de portos como saída ou entrada através do uso de máscaras de *bits*.

A comutação é assegurada pelo estabelecimento de um perído de tempo fixo para a captação de uma antena, ou seja, cada configuração $A0$ e $A1$ assume um determinado valor da tabela durante este período de tempo até assumir o próximo valor, alterando assim a antena a captar. Utilizando um dos *timers* disponíveis no microcontrolador MSP430 é possível definir uma taxa de comutação de 83.3 kHz. Quando um período temporal especificado é atingido no temporizador é gerada uma interrupção à qual está associada uma rotina responsável pela análise do estado atual e da passagem para o estado seguinte.

4.5 Componentes de *software* e *hardware* da plataforma de aquisição de dados

A plataforma de aquisição de dados baseada na placa *Raspberry Pi Model B+* necessita da instalação de um sistema operativo e do pacote de *software rtlsdr* para desempenhar a sua função dentro do projeto em implementação.

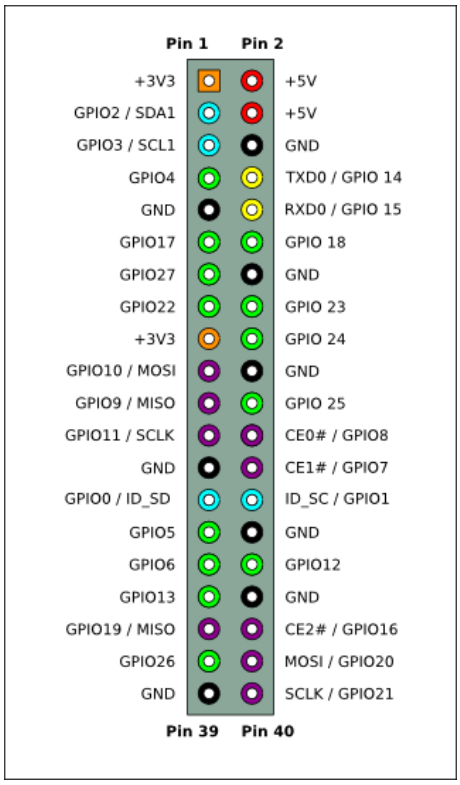

Figura 4.12: Esquema representativo dos pinos de uso geral do *Raspberry Pi Model B+*

O sistema operativo escolhido para o *Raspberry Pi Model B+* denomina-se *Raspbian* [23] e é uma implementação baseada no sistema operativo *Debian* [46] especialmente otimizada para o *hardware* presente em computadores *Raspberry Pi*. Para instalar o sistema operativo no *Raspberry Pi Model B+* utilizou-se um cartão de memória SD, visto que o *Raspberry Pi Model B+* não possui memória de armazenamento próprio.

4.5.1 Utilização de GPIO

O *Raspberry Pi Model B+* possui pinos designados como *general purpose input-output ports* (GPIO). Estes portos podem ser usados como entradas ou saídas digitais. O esquema 4.12 permite identificar a função de cada um dos pinos.

No âmbito do presente projeto, o GPIO será utilizado para alimentar a placa de desenvolvimento MSP-EXP430G2 *Launchpad*, visto que esta *devboard* pode ser alimentada a 3.3 V (ver secção 3.2.3) e o *Raspberry Pi Model B+* possui essa opção nos seus portos GPIO. Para a utilização dos pinos de alimentação, isto é, o pino de 3.3 V e o pino de *ground* não foi necessário desenvolver nenhum tipo de *software*.

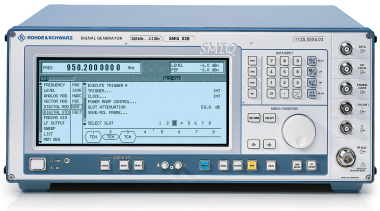

Figura 4.13: Gerador de sinais SMIQ03B do fabricante *Rhodes & Schwarz*

4.6 *Timestamping* para sincronização

A identificação de cada antena presente no *array* comutado deve ser efetuada de forma síncrona, uma vez que a captura de dados é executada sem informação sobre qual antena se encontra a captar. Ora, como existe um *dongle RTL* com uma única entrada para quatro canais comutados, é crucial utilizar alguma forma de *timestamping* nos dados para que seja possível recuperar a informação temporal sobre cada canal.

A solução encontrada para este problema consiste na utilização de um pino do MSP-EXP430G2 *Launchpad* a sinalizar a captura por parte de uma antena específica. Atribui-se um valor de "1"do pino como sinalizando a captura por parte da antena 1, ficando o valor "0"com o significado de não seleção da antena 1. Essa informação é disponibilizada a um equipamento de sinalização que modula a informação numa portadora de frequência intermédia, cujo valor de frequência cai dentro da banda de saída do sintonizador R820T. O sinal modulado é injetado diretamente no *dongle RTL* no caminho em fase (I) ou em quadratura (Q) entre o sintonizador e o conversor analógico-digital RTL2832u, sendo posteriormente amostrado e quantizado em valor digital juntamente com os sinais capturados. Conhecidos os instantes temporais em que a antena 1 está a captar, através da filtragem do sinal de *timestamping*, são facilmente calculados os outros canais, uma vez que a frequência de comutação entre as antenas é conhecida.

Um aspeto a salientar é o facto de esta operação de injeção direta de *timestamping* no *dongle RTL* requerer a conversão entre sinal não-balanceado e sinal balanceado, visto que os caminhos do sintonizador para o ADC do *dongle RTL* serem diferenciais. Para ultrapassar este obstáculo, foi utilizado um *balun* com razão de impedâncias 4:1, uma vez que as linhas de transmissão do *dongle RTL* apresentam uma impedância próxima de 200 Ω e a saída do equipamento de sinalização encontra-se adaptada a 50 Ω.

4.6.1 Equipamento de sinalização

Por forma a gerar uma forma de onda de sincronização para posterior identificação dos canais representando cada uma das antenas do *array*, foi utilizado o gerador de sinais SMIQ03B do fabricante *Rhodes & Schwarz* [47]. Este equipamento consegue gerar sinais dentro de uma banda de frequências que vai desde os 300 kHz até aos 6.4 GHz com potências de sinal que podem chegar até aos 16 dB mW.

4.6.2 *Hacking* do *dongle RTL*

A injeção direta do sinal de *timestamping* entre o sintonizador e o conversor analógico-digital requer uma pequena cirurgia à placa de circuito impresso do *dongle RTL*. Para tal, é necessário seguir os seguintes passos:

- Identificar um caminho diferencial I ou Q entre o sintonizador R820T e o ADC
- Remover o isolamento de cada uma das pistas diferenciais para expor o material condutor
- Soldar um fio condutor em cada pista exposta
- Soldar ao balun a extremidade livre de cada um dos fios conetados às pistas
- Conetar ao balun o equipamento de sinalização

4.7 Comunicação entre plataformas de aquisição e processamento de dados

Como referido anteriormente 3.2.5, o *software rtl-sdr* [28] possui componentes de grande valor para o exercício em prática neste projeto. Uma delas é o programa *rtl_tcp* cuja funcionalidade é a de criar um servidor de dados I/Q em *streaming* [48] para que qualquer aplicação consiga receber estes dados através de TCP/IP.

Para potenciar o uso da ferramenta *rtl_tcp*, é necessário desenvolver uma forma de realizar a conexão remota ao servidor de dados e armazená-los para posterior processamento. O código desenvolvido para executar esta tarefa foi realizada em linguagem *Matlab*, tirando partido das funções de conexão *tcpip* [49] presentes no pacote *Instrument Control Toolbox* [50].

4.7.1 *Sockets TCP/IP*

A aplicação desenvolvida para a comunicação entre as plataformas de aquisição (*Raspberry Pi Model B+*) e processamento de dados utiliza um mecanismo de estabelecimento de ligação e transferência de dados baseado em *sockets* TCP/IP. Um *socket* é uma abstração que disponibiliza a uma aplicação a capacidade de envio e receção de dados da mesma forma que um descritor de ficheiro possibilita a escrita e leitura de ficheiros armazenados em disco [51]. No entanto, um *socket* permite que uma aplicação ligada a uma determinada rede troque informação com uma outra aplicação que esteja conetada à mesma rede. Este mecanismo pode ser visto como um canal no qual uma das aplicações envia dados por uma extremidade e a aplicação na outra extremidade do canal recebe esses mesmos dados, acontecendo o mesmo no sentido contrário.

O código desenvolvido para esta aplicação está presente em A.

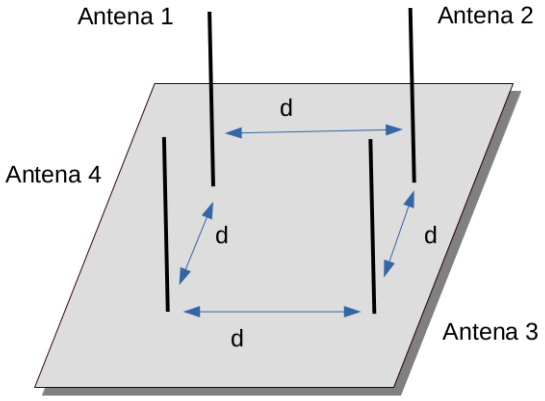

Figura 4.14: *Array* de antenas espaçadas por distância d

4.8 Antenas

Cada uma das antenas escolhidas para fazer a aquisição de dados dos sinais RF de interesse deve assumir uma posição fixa relativamente às outras, sendo que esta propriedade deve ser comum a todas as antenas. O interesse desta propriedade deve-se ao facto de ser importante conhecer a distância entre cada uma das antenas para que a computação da diferença de fase entre as antenas seja possível.

O esquema da figura 4.14 representa a disposição das antenas relativamente ao plano e a cada uma delas. O espaçamento d entre cada antena vizinha será dependente do comprimento de onda λ do sinal de interesse, o que por sua vez, é calculado a partir da frequência do sinal através da relação $\lambda = \frac{c}{f}$, onde c representa a constante de velocidade da luz com o valor $300\,000\,\mathrm{km\,s^{-1}}$ e f é a frequência do sinal de interesse. A distância d entre cada antena vizinha é então estipulada como:

$$d = \frac{\lambda}{4} \tag{4.1}$$

Desta forma, a separação entre cada antena vizinha em fase corresponde a $\pi/2$ ou a $90°$.

Esta estrutura foi montada numa placa de esferovite devido à facilidade de obtenção do material na Faculdade de Engenharia.

4.9 Implementação dos blocos de processamento de sinal

O processamento de sinal requerido para este projeto foi levado a cabo utilizando o *software Matlab* para realizar cada um dos blocos referidos em 3.3. A implementação de cada um dos blocos é documentada nas seguintes subsecções.

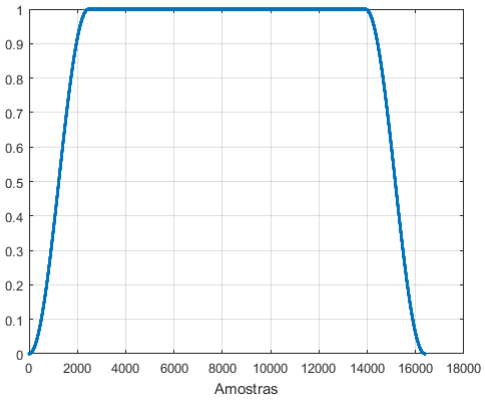

Figura 4.15: Janela de *Tukey*

4.9.1 Leitura de dados

Os dados utilizados para computar a direção de chegada de um sinal incidente foram adquiridos através da gravação de amostras no *software Airspy SDRSharp*. Como tal, as amostras encontram-se em formato *WAVE*.

O *software* Matlab contém uma função denominada *wavread* [52] que permite a importação de amostras em formato *WAVE* para o ambiente de desenvolvimento. Como tal, este foi o procedimento usado para adquirir os dados relevantes para o processamento de dados de forma a permitir a manipulação dos mesmos.

Para elaborar o processamento de forma eficiente, foi selecionado um número fixo para cada bloco de amostras. Cada bloco incorpora 16384 (2^{14}) amostras. A decisão do número de amostras por bloco foi tomada tendo em conta o melhoramento de desempenho do algoritmo FFT com blocos de comprimentos par relativamente a blocos de comprimento ímpar.

Após a seleção dos dados relevantes, é aplicada uma janela de *Tukey* aos dados. A janela de *Tukey* aplicada está representada na figura 4.15. Esta janela é concebida através do comando Matlab *tukeywin* [53] e serve para impedir a ocorrência de um fenómeno conhecido como *spectral leakage* [54].

4.9.2 Projeto de filtros

Os filtros implementados no âmbito do processamento digital de sinal do presente projeto derivaram da utilização dos comandos Matlab *fir1* [55], com a sua aplicação a ser efetuada através do comando *filter* [56].

4.9.3 *Demultiplexing* de canais

A desmultiplexagem dos canais presentes nas amostras pressupõe os passos enumerados em 3.3.3. A implementação de cada um dos procedimentos é explicado em seguida.

A geração local da onda quadrada de referência é alcançada através da utilização do comando *square* [57]. Sabendo o período de cada impulso utilizado para sinalizar a captura de uma determinada antena, é possível gerar esta referência à frequência estipulada para o *timestamping*. O sinal gerado terá valores de amplitude 0 ou 1.

A correlação entre a onda de referência e os dados capturados é feita através da multiplicação da FFT das amostras com o sinal de referência. Para realizar a FFT das amostras é utilizada a função *fft* [58] presente no Matlab.

A determinação do atraso entre a onda de referência e o sinal de sincronismo presente nos dados executa-se através da computação do valor de amplitude máxima do módulo dos dados da correlação feita anteriormente. A amostra à qual se apresenta esse valor de amplitude representa o atraso entre a onda de referência e o sinal de sincronização presente nos dados.

A geração dos sinais seletores pode ser elaborada através da chamada à função *square* com os mesmos parâmetros da onda de referência, no que toca à amplitude e frequência. No entanto, cada um dos quatro seletores de canal terão uma diferença de fase de $\pi/2$ entre cada um dos canais consecutivos, tendo também em conta o atraso inicial computado anteriormente.

Por fim, é realizada a multiplicação entre cada um dos sinais seletores pelos dados relativos ao sinal de interesse. Desta forma, as amostras de cada antena ficam discriminadas.

4.9.4 Procedimento de decisão de direção

O procedimento de decisão de direção não foi alvo de implementação prática ao longo da elaboração da presente dissertação.

4.9.5 Visualização da direção de chegada

A funcionalidade de visualização da direção de chegada não foi implementada no decurso do presente projeto.

4.10 Conclusões

Devido à existência de material pertinente para a construção do sistema nas imediações da Faculdade de Engenharia, a implementação ao nível dos módulos físicos, apesar de custosa em algumas partes, realizou-se de forma eficiente. Também foi possível desenvolver os componentes de *software* eficientemente devido à documentação disponível *online*.

Capítulo 5

Desempenho do Sistema

5.1 Introdução

O presente capítulo demonstra alguns resultados do sistema que se consideram pertinentes. São elaborados casos de teste com especificações ao nível da frequência de receção, da frequência de amostragem e da distância entre emissor e recetor, considerando que o emissor é conhecido em termos de frequência e de posição.

5.2 Critérios e estipulação de testes

O sistema de localização implementado apenas pode ser qualificado como funcional na ocorrência do correto desempenho dentro dos casos de uso que se estipulam como sendo os essenciais. Para obter resultados com um determinado nível de granularidade e para permitir uma análise exata, o correto desempenho será classificado como tal no caso de cumprimento com um conjunto de critérios. Como tal, revela-se oportuno referir quais são os critérios adotados para efetuar a análise de desempenho do sistema:

- Capacidade de determinação da direção
- Sensibilidade a variações de posição
- Alcance do sistema
- Sensibilidade a obstáculos

Tendo estabelecido o conjunto de critérios para a análise da *performance* do sistema, é tempo para definir os casos de uso que devem ser testados:

- Teste com alvo estático
- Teste com alvo móvel

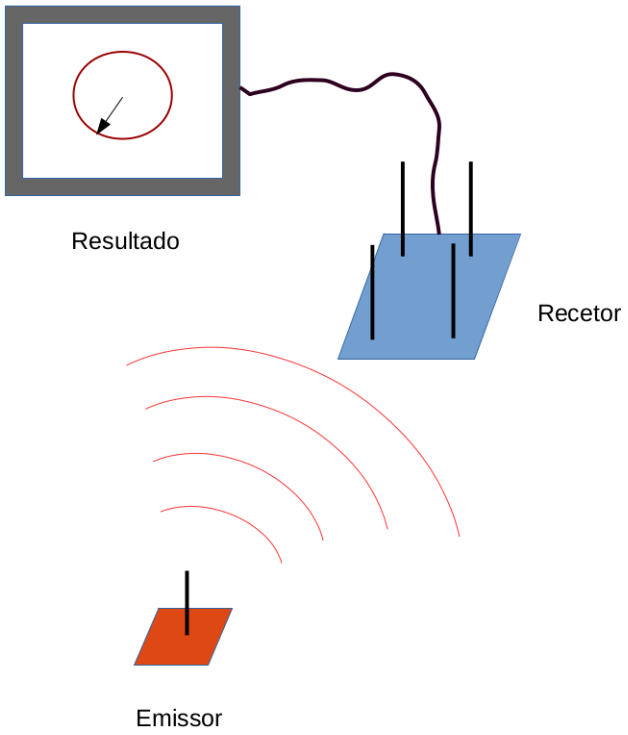

Figura 5.1: Caso de uso com alvo estático e resultado

No caso do teste com alvo estático, o recetor deverá conseguir obter e indicar direção de um emissor que se encontre fixo num determinado ponto espacial conhecido previamente. O conhecimento da posição do emissor em antemão permite confrontar os resultados obtidos pelo sistema e a direção real do emissor.

Na situação de análise do caso de uso com um emissor móvel, o recetor deverá indicar corretamente a direção do emissor e deverá ser capaz de seguir o emissor no caso de este se deslocar numa direção diferente relativamente ao recetor.

Ambas as situações de teste presumem que o recetor se encontra estático.

5.3 Teste com alvo estático

A situação de teste com alvo estático encontra-se representada no diagrama 5.1. No diagrama, assume-se que o emissor se encontra estabilizado numa determinada posição espacial, assim como o recetor. Após a receção do sinal emitido pelo emissor, o recetor deve recebê-lo e encaminhá-lo para a plataforma de processamento de dados, demonstrando em seguida os resultados. A situação encontra-se ilustrada no diagrama.

Este caso de uso será testado para a frequência de emissão de 144.550 MHz, frequência central de sintonização do *dongle RTL* de 143.950 MHz e sinal de sincronização à frequência de

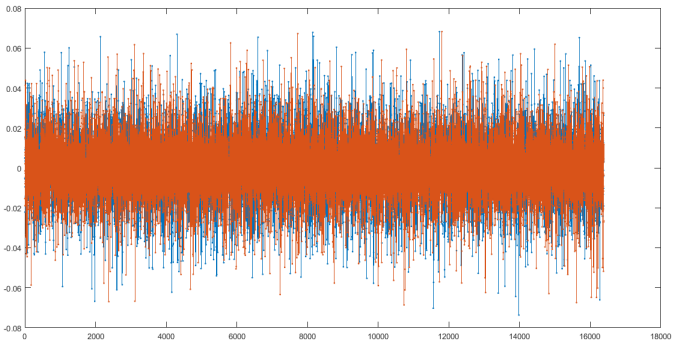

Figura 5.2: Dados capturados - domínio temporal

4.170 MHz. A taxa de amostragem é de 2.4 milhões de amostras por segundo. O *switch* de antenas está a ser comutado a uma taxa de 33 kHz.

A captura de dados é elaborada através do *software Airspy SDRSharp*. As amostras são importadas para o ambiente Matlab, sendo selecionadas apenas as amostras representativas de um bloco de dados, ou seja, 16384 amostras. A figura 5.2 representa os valores das amostras, onde a azul é representada a parte real (I) e a laranja é representada a parte imaginária (Q). A figura 5.3 representa o espetro das amostras recolhidas normalizado pelo número de amostras do bloco de dados, centrado à frequência zero.

O sinal mais forte à direita da frequência zero representa o sinal incidente no *array* de antenas. O sinal composto por vários harmónicos que se situa à esquerda da frequência zero representa o sinal de sincronização.

Após a importação dos dados e a sua visualização, é feita a exclusão dos componentes que não oferecem nada de relevante ao processamento a ser efetuado. Após a filtragem passa-banda dos componentes de sinal incidente e sinal de sincronismo, o espetro dos dados é apresentado na figura 5.4.

O sinal de sincronismo é então isolado e deslocado para frequência zero por forma a computar-se o seu atraso relativamente a uma onda de referência. O espetro do sinal é representado em 5.5 e pode ser visto em detalhe na figura 5.6.

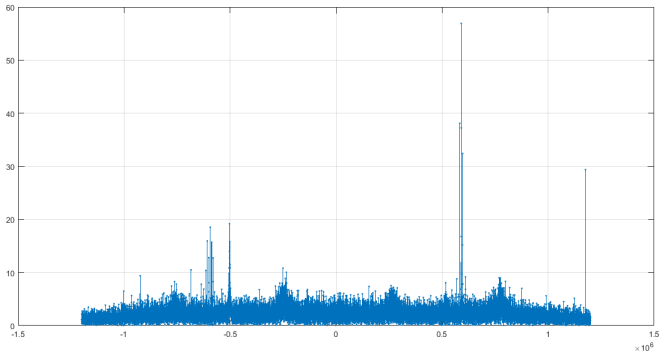

Figura 5.3: Dados capturados - domínio de frequência

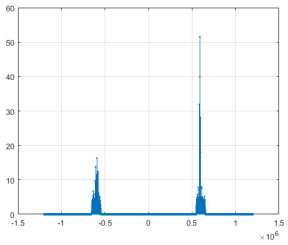

Figura 5.4: Dados filtrados (sinal e *timestamp* - domínio de frequência

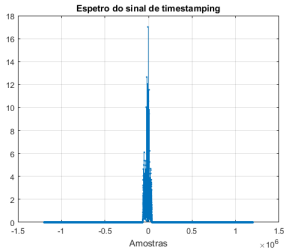

Figura 5.5: Sinal de sincronização deslocado para a frequência zero

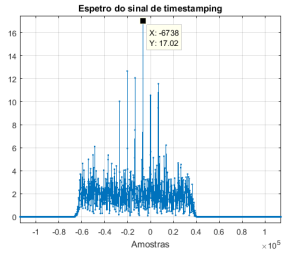

Figura 5.6: Pormenor do sinal de sincronismo

A geração local da onda de referência para correlacionar com os dados recolhidos dá origem ao sinal presente na figura 5.7.

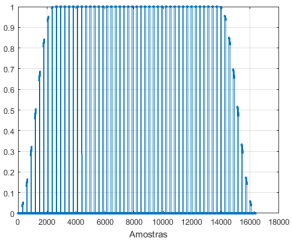

Figura 5.7: Sinal de referência para selecionar antenas

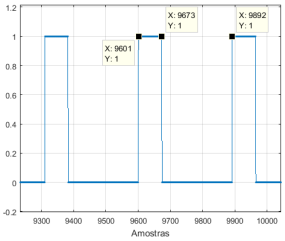

Figura 5.8: Pormenor do sinal de referência

Após a filtragem e multiplicação dos sinais seletores pelas amostras do sinal incidente, os canais são selecionados. A representação temporal de cada um dos canais está demonstrada nas figuras 5.9 e 5.10. Pode ver-se que cada um dos sinais está discriminado de acordo com a antena que o capta.

Não é possível obter resultados relativamente à direção do sinal devido à falta de implementação dos módulos de *software* referentes a essas tarefas.

5.4 Teste com alvo em movimento

O caso de uso para o qual o emissor se encontra em movimento relativamente ao recetor está ilustrado no diagrama 5.11. Aqui será testada a capacidade do sistema responder a mudanças de direção do emissor. O conhecimento do percurso efetuado pelo emissor está assegurado, uma vez

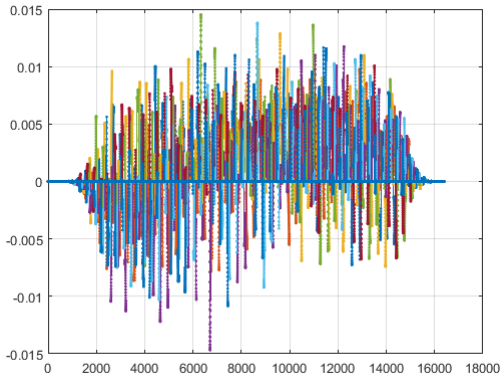

Figura 5.9: Dados de cada uma das antenas

que será necessário conhecer o percurso e as inerentes mudanças de direção para que os resultados do sistema de localização sejam confrontados com os resultados reais.

O caso de uso em questão não foi testado no âmbito da presente dissertação.

5.5 Conclusão

O presente capítulo serviu o propósito de documentar a análise de desempenho do sistema, elaborando situações dentro dos casos de uso estipulados como sendo os mais importantes para o correto funcionamento do sistema. Infelizmente, como algumas das funcionalidades do sistema não foram implementadas, não foi possível fazer uma análise detalhada da performance do sistema.

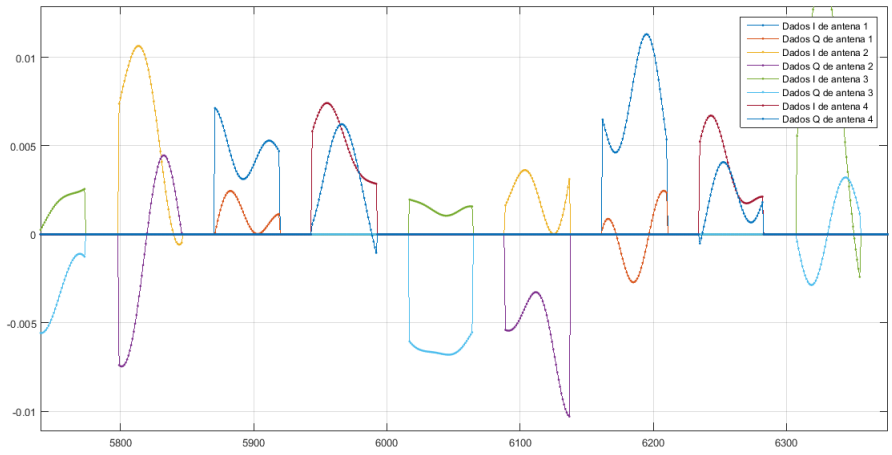

Figura 5.10: Pormenor do sinal de interesse

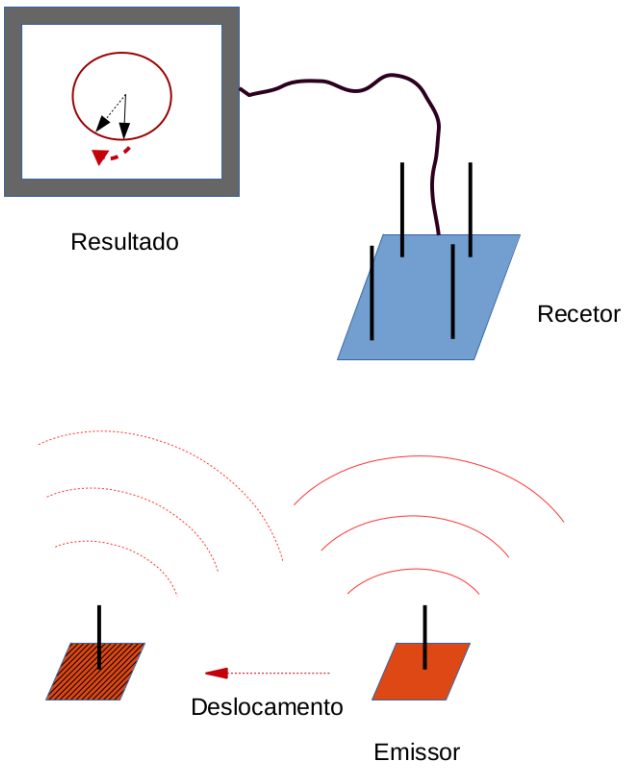

Figura 5.11: Caso de uso com alvo móvel e resultado

Capítulo 6

Conclusões

O propósito deste capítulo é o de fornecer o leitor com as conclusões relevantes obtidas através da execução do projeto de sistema de localização rádio via *software-defined radio*.

6.1 Conclusões gerais

O projeto em análise no presente documento foi planeado como sendo uma tentativa de utilizar um equipamento *low cost* e *off-the-shelves* como base de aquisição de sinal para posterior processamento e análise da direção de um determinado emissor. A metodologia escolhida para alcançar o objetivo deste trabalho seguiu uma veia *do it yourself* de rápida prototipagem seguida de *deployment*.

O principal obstáculo da realização deste trabalho centrou-se nos defeitos do *dongle RTL*. Este equipamento serve para realizar captura de sinais de televisão digital terrestre e foi concebido com esse objetivo em mente. Sendo um dispositivo de aplicação específica, a alteração do seu comportamento é uma tarefa que requer experimentação ao longo de várias iterações. Como tal, o facto de não existir documentação disponível sobre conversor RTL2832 e sobre o *layout* da placa de circuito impresso tornaram a tarefa do presente projeto mais desafiante.

O objetivo da realização de um sistema de localização com a representação de direção de emissor não foi cumprido. A integração com os vários tipos de comunicação referidas na introdução deste documento também não foi efetuada, assim como a verificação do caso de uso principal, que seria a integração do sistema num veículo automóvel. A falta de planeamento e organização são identificadas como as principais causas para a incompleta conclusão deste trabalho.

6.2 Trabalho futuro

O tipo de sistema que se tentou projetar e implementar ao longo deste trabalho necessitaria de várias iterações até estar na sua forma mais eficiente. Para atingir esse objetivo, um possível trabalho futuro de um projeto no âmbito do mesmo tema de dissertação deveria focar-se nos seguintes pontos:

- Desenvolvimento e implementação de algoritmos para cálculo de fase e computação de direção
- Criação da *graphical user interface*
- Instalação do sistema no teto de um automóvel
- Integração do sistema com um automóvel
- Alternativa ao equipamento de sinalização utilizado para tornar o sistema portátil
- Desenho de uma caixa para enclausurar o sistema
- Alternativa ao sistema de sincronização baseado em *timestamping*
- Derivação de um algoritmo de processamento alternativo
- Tornar o projeto exclusivamente implementado em ferramentas *open source* e disponibilizá-lo à comunidade *online*

6.3 Outras abordagens possíveis

A metodologia e o projeto de sistema utilizados para desenvolver este sistema de localização baseado em *software-defined radio* não são as únicas possíveis para tentar alcançar o objetivo estipulado. Segue uma lista de alternativas à metodologia e projetos seguidos ao longo da execução deste projeto:

- Utilização de quatro *dongles RTL* com um *clock* comum para efetuar receção múltipla de forma coerente
- Determinação do *jitter* introduzido pela saída *USB* do *dongle RTL* e implementação do método de sincronização de dados através de *software*
- Centrar a aquisição e processamento de dados numa única plataforma e envio de dados apenas para visualização. Desta forma, seria possível a visualização dos dados em qualquer tipo de plataforma *Android* ou *iOS*.

Anexo A

Código desenvolvido

A.1 Código do microcontrolador MSP430

```
#include <msp430g2553.h>

/*
        P1.4 -> A0
        P1.5 -> A1
        P1.7 -> ~EN
        P2.0 -> INT1
        P2.1 -> INT2
        P2.3 -> INT3
        P2.4 -> INT4
*/

int main(void)
{
        WDTCTL = WDTPW + WDTHOLD; // Prevent unwanted resets by
            stopping the watchdog timer

        //Timer initialization
        CCTL0 = CCIE;
        TACTL = TASSEL_2 + ID_0 + MC_1;
        CCR0 = 12;

        P1OUT &= 0x00;
        P1DIR &= 0x00;
        P1DIR |= BIT4 + BIT5 + BIT7; //Outputs
```

```
        //P1OUT &= ~BIT7; //enabler set to low level in order to
            enable the antenna switch
        //P1OUT &= ~BIT4 + ~BIT5; //A0 = 0 and A1 = 0 to switch
           to RF1
        P1OUT = 0x00; //A0 = 0 and A1 = 0 to switch to RF1

        P2DIR |= BIT0 + BIT1 + BIT2 + BIT3;
        P2OUT &= 0x01; //All pins on low level except bit0 which
            sinalizes RF1

        BIS_SR(CPUOFF + GIE);

        for(;;)
        {
        }
}

#pragma vector = TIMERA0_VECTOR
__interrupt void CCR0_ISR(void)
{
        //do switching
                switch(P1OUT)
                {
                        case 0x00:
                                //switch to RF2
                                P1OUT = 0x10;
                                //Generate high level on P2.1
                                P2OUT = 0x02;
                                break;
                        case 0x10:
                                //switch to RF3
                                P1OUT = 0x20;
                                //Generate high level on P2.2
                                P2OUT = 0x04;
                                break;
                        case 0x20:
                                //switch to RF4
                                P1OUT = 0x30;
                                //Generate high level on P2.3
                                P2OUT = 0x10;
```

```
                    break;
                case 0x30:
                    //switch to RF1
                    P1OUT = 0x00;
                    //Generate high level on P2.0
                    P2OUT = 0x01;
                    break;
            }
}
```

A.2 Código de processamento

```
function [ corr_data ] = correlationWithTimestamp(
    complex_samples, smoothed_wave )

corr_data = fft(complex_samples).*smoothed_wave;
end

function [ x_square ] = generateSquareWave( f_square, fs,
    phase_shift, duty_cycle, L )

t = 0:1/fs:(L-1)/fs;
x_square = square(2*pi*t*f_square+phase_shift, duty_cycle);
x_square = x_square./2 + 1/2;

end

function [ shifted_data ] = shiftToOrigin( complex_samples, fs,
    shift_freq )

L = length(complex_samples);
shifted_data = complex_samples.*exp(2*pi*j*(0:L-1)'/fs*
    shift_freq);

end

function [ smoothed_data ] = smooth_data_Tukey( complex_samples
    )

L = length(complex_samples);
w = tukeywin(L, 0.3);
```

```
smoothed_data = complex_samples.*w;

end

function [ filteredData ] = timestamping_filtering(complex_data,
    Wn)
%TIMESTAMPING_FILTERING Function used to filter a specific
    spectral
%component of a signal
%
b_lowpass = fir1(1024, Wn, 'low');
filteredData = filter(b_lowpass, 1, complex_data);
end

clear all;
close all;


[data, fs] = wavread('C:\sdr-sharp\captura_5fev_baofeng.wav');
bufferLength = 2^14;
nCycles = length(data)/bufferLength;

%for n = 0:nCycles-1


    samples = data(1:2^14,:)*[1;j];
    L = length(samples);

    %smoothing by Tukey window
    smoothed_data = smooth_data_Tukey(samples);

    %passband filter timestamp and signal
    b_pass = fir1(1024, [(5.5e5)/(fs/2) (6.5e5)/(fs/2)]);
    filtered_data = filter(b_pass, 1, smoothed_data);

    %isolate timestamp
    filtered_fft = fft(filtered_data);
    timestamping_fft = filtered_fft;
    timestamping_fft(1:L/2) = 0;
    timestamping_data = ifft(timestamping_fft);
```

```
%find peak
[~, I] = max(abs(timestamping_fft));

%Shifting timestamping to origin
shifted_data = shiftToOrigin(timestamping_data, fs, I);

%Square wave generation
f_square = 33e3/4;
x_square = generateSquareWave(f_square, fs, 0, 25, L);

%Square wave smoothing
x_square_smoothed = smooth_data_Tukey(x_square');

%Signal cross correlation with square wave
corr_data = correlationWithSquareWave(shifted_data,
    x_square_smoothed);

%Find cross correlation peak
[M2, I2] = max(real(corr_data));
I2
I2 = mod(I2, round(fs/33e3)-1);

%Signal data extraction
signal_fft = filtered_fft;
signal_fft(L/2:L) = 0;
signal = ifft(signal_fft);

%Signal data shift
signal_data = shiftToOrigin(signal, fs, -5.9e5);

signal_data = smooth_data_Tukey(signal_data);

%tukey window
w = tukeywin(L, 0.3);

%Antena 1
ant1_selector = generateSquareWave(f_square, fs, I2, (2/3)
    *25, L);
ant1_selector_smooth = ant1_selector'.*w;
```

```
ant1_data = signal_data.*ant1_selector_smooth;

%Antena 2
ant2_selector = generateSquareWave(f_square, fs, I2+pi/2,
    (2/3)*25, L);

ant2_selector_smooth = ant2_selector'.*w;

ant2_data = signal_data.*ant2_selector_smooth;

%Antena 3
ant3_selector = generateSquareWave(f_square, fs, I2+pi,
    (2/3)*25, L);

ant3_selector_smooth = ant3_selector'.*w;

ant3_data = signal_data.*ant3_selector_smooth;

%Antena 4
ant4_selector = generateSquareWave(f_square, fs, I2+(3/2)*pi
    , (2/3)*25, L);

ant4_selector_smooth = ant4_selector'.*w;

ant4_data = signal_data.*ant4_selector_smooth;

%DOA Estimation
corr_1_2 = conj(fft(ant1_data)).*(fft(ant2_data));
corr_3_4 = conj(fft(ant3_data)).*fft(ant4_data);
```

Referências

[1] G. Pont, L. Rolo, e S. Cunha. The STRAtospheric PLatform EXperiment (STRAPLEX) A low-cost solution for near-space experiments. Em *36th COSPAR Scientific Assembly*, volume 36 de *COSPAR Meeting*, página 354, 2006.

[2] Andrea Corte-Real. Deorbiting and reentry of gamasat, 2012.

[3] Marco Luise Davide Dardari, Emanuela Falletti. *Satellite and Terrestrial Radio Positioning Techniques, A Signal Processing Perspective*. Elsevier, 2012.

[4] Cheltek Defence. Direction finding and position fixing techniques.

[5] R. O. Schmidt. A new approach to geometry of range difference location. *IEEE Transactions on Aerospace and Electronic Systems*, 1972.

[6] R. O. Schmidt. Least squares range difference location. *IEEE Transactions on Aerospace and Electronic Systems*, 1996.

[7] Lin K. Liu B. On the accuracy analysis of the distance-difference estimation for sssd positioning method in wireless communications. *IEEE International Conference on Communications*, 2007.

[8] Phillippe Crumeyrolle. Source localization by conjugated delays method main description of a new process of passive localization. *American Journal of Signal Processing*, 2012.

[9] RJ LACKEY e DW UPMAL. SPEAKEASY - THE MILITARY SOFTWARE RADIO. *IEEE COMMUNICATIONS MAGAZINE*, 33(5):56–61, MAY 1995. doi:10.1109/35.392998.

[10] Jose Raul Machado-Fernandez. Software Defined Radio: Basic Principles and Applications. *Facultad de Ingeniera* , 24:79 – 96, 01 2015. URL: http://www.scielo.org.co/scielo.php?script=sci_arttext&pid=S0121-11292015000100007&nrm=iso.

[11] John B. Anderson e Rolf Johnnesson. *Understanding Information Transmission (IEEE Press Understanding Science & Technology Series)*. Wiley-IEEE Press, 2005.

[12] Spectrum IEEE Stephen Cass. A $40 software-defined radio, Julho 2015. URL: http://spectrum.ieee.org/geek-life/hands-on/a-40-softwaredefined-radio.

[13] Realtek. Página oficial realtek, Julho 2015. URL: http://www.realtek.com.tw/products/productsView.aspx?Langid=1&PFid=35&Level=4&Conn=3&ProdID=257.

[14] Yaesu. Welcome to yaesu.com, Janeiro 2016. URL: http://www.yaesu.com/indexVS.cfm?cmd=DisplayProducts&ProdCatID=111&encProdID=5CB596EBED9A3EE26635C7E1F02500D9.

[15] Analog Devices. *Wideband 2.5 GHz, 37 dB Isolation at 1 GHz, CMOS 1.65 V to 2.75 V, 4:1 Mux/SP4T*, 8 2013. Rev. C.

[16] Texas Instruments. *MSP-EXP430G2 LaunchPad Experimenter Board - User's Guide*, 7 2012.

[17] Texas Instruments. Msp430g2553 | msp430g2x/i2x | ultra-low power | description & parametrics, Dezembro 2015. URL: http://www.ti.com/product/msp430g2553.

[18] Texas Instruments. Code composer studio (ccs) integrated development environment (ide) - ccstudio - ti tool folder, Dezembro 2015. URL: http://www.ti.com/tool/ccstudio.

[19] Energia. Energia, Dezembro 2015. URL: http://energia.nu/.

[20] Rafael Microelectronics, Inc. *R820T High Performance Low Power Advanced Digital TV Silicon Tuner Datasheet*, 11 2011. Rev 1.2.

[21] Antti Palosaari. Antti's linuxtv blog: Naked hardware #11: unbranded rtl2832u + r820t, Dezembro 2015. URL: http://blog.palosaari.fi/2013/06/naked-hardware-11-unbranded-rtl2832u.html.

[22] eLinux.org. Rpi hardware, Dezembro 2015. URL: http://elinux.org/RPi_Hardware.

[23] Raspbian. Raspbian, Dezembro 2015. URL: https://www.raspbian.org/.

[24] steve m. steve-m/librtlsdr - github, Dezembro 2015. URL: https://github.com/steve-m/librtlsdr.

[25] osmocomSDR. rtl-sdr - osmocomsdr, Dezembro 2015. URL: http://sdr.osmocom.org/trac/wiki/rtl-sdr.

[26] Raspberry Pi Foundation. Raspberry pi - index page, Dezembro 2015. URL: https://www.raspberrypi.org/forums/.

[27] Raspberry Pi Foundation. Raspberry pi documentation, Dezembro 2015. URL: https://www.raspberrypi.org/documentation/.

[28] keenerd. keenerd/rtl-sdr - github, Dezembro 2015. URL: https://github.com/keenerd/rtl-sdr.

[29] Raspberry Pi Foundation. Power supply - raspberry pi documentation, Dezembro 2015. URL: https://www.raspberrypi.org/documentation/hardware/raspberrypi/power/README.md.

[30] rtl sdr.com. Quick start guide - rtl-sdr.com, Dezembro 2015. URL: http://www.rtl-sdr.com/rtl-sdr-quick-start-guide/.

[31] Mathworks. Matlab - the language of technical computing, Dezembro 2015. URL: http://www.mathworks.com/products/matlab/.

[32] Mathworks. Digital signal processing (dsp) - matlab & simulink solutions, Dezembro 2015. URL: http://www.mathworks.com/solutions/dsp/.

[33] Mathworks. Code examples - matlab, Dezembro 2015. URL: http://www.mathworks.com/products/matlab/examples.html.

[34] GNU Radio e Qt. Gqrx sdr – a software defined radio powered by gnu-radio and qt, Janeiro 2016. URL: http://gqrx.dk/.

[35] Airspy. Airspy sdr# | low cost high performance software defined radio, Janeiro 2016. URL: http://airspy.com/.

[36] rtl sdr.com. About rtl-sdr - rtl-sdr.com, Janeiro 2016. URL: http://www.rtl-sdr.com/about-rtl-sdr/.

[37] Prof. Peter Kabal. Wave file specifications, Janeiro 2016. URL: http://www-mmsp.ece.mcgill.ca/Documents/AudioFormats/WAVE/WAVE.html.

[38] Mathworks. Floating-point numbers - matlab & simulink, Janeiro 2016. URL: http://www.mathworks.com/help/matlab/matlab_prog/floating-point-numbers.html.

[39] Barry Van Veen Simon Haykin. *Signals and Systems*. John Wiley & Sons, Inc., 2003.

[40] Kicad. Kicad eda, Dezembro 2015. URL: http://kicad-pcb.org/about/kicad/.

[41] Raspberry Pi Foundation. Gpio - raspberry pi documentation, Dezembro 2015. URL: https://www.raspberrypi.org/documentation/hardware/raspberrypi/gpio/README.md.

[42] Microchip. *250 mA, 16V, Low Quiescent Current LDO Regulator*, 2011.

[43] IEEE. Ieee p802.3af dte power via mdi task force, Dezembro 2015. URL: http://www.ieee802.org/3/af/.

[44] pinoutsguide.com. Power over ethernet (poe) pinout diagram, Dezembro 2015. URL: http://pinoutsguide.com/Net/poe_pinout.shtml.

[45] Traco Power. *DC/DC Converters TSR-3 Series, 3 A*, Março 2014. URL: www.tracopower.com.

[46] Debian. Debian – o sistema operacional universal, Janeiro 2016. URL: https://www.debian.org/.

[47] Rohde & Schwarz. R&s®smiq vector signal generator - overview - rohde & schwarz united states, Janeiro 2016. URL: https://www.rohde-schwarz.com/us/product/smiq-productstartpage_63493-7561.html.

[48] Ubuntu Manpage. Ubuntu manpage: rtl-tcp - an i/q spectrum server for rtl2832 based dvb-t receivers, Janeiro 2016. URL: http://manpages.ubuntu.com/manpages/trusty/man1/rtl_tcp.1.html.

[49] Mathworks. Create tcpip object - matlab tcpip, Janeiro 2016. URL: http://www.mathworks.com/help/instrument/tcpip.html.

[50] Mathworks. Instrument control toolbox documentation, Janeiro 2016. URL: http://www.mathworks.com/help/instrument/index.html.

[51] Kenneth L. Calvert Michale J. Donahoo. *TCP/IP Sockets In C - Practical Guide For Programmers - Second Edition*. Elsevier, 2009.

[52] Mathworks. Read wave (.wav) sound file - matlab wavread, Janeiro 2016. URL: http://www.mathworks.com/help/matlab/ref/wavread.html.

[53] Mathworks. Tukey (tapered cosine) window - matlab tukeywin, Janeiro 2016. URL: http://www.mathworks.com/help/signal/ref/tukeywin.html.

[54] National Instruments. Understanding ffts and windowing, Janeiro 2016. URL: http://www.ni.com/white-paper/4844/en/#toc2.

[55] Mathworks. Window-based fir filter design - matlab fir1, Janeiro 2016. URL: http://www.mathworks.com/help/signal/ref/fir1.html.

[56] Mathworks. 1-d digital filter - matlab filter, Janeiro 2016. URL: http://www.mathworks.com/help/matlab/ref/filter.html.

[57] Mathworks. Square wave - matlab square, Janeiro 2016. URL: http://www.mathworks.com/help/signal/ref/square.html.

[58] Mathworks. Fast fourier transform - matlab fft, Janeiro 2016. URL: http://www.mathworks.com/help/matlab/ref/fft.html.

[59] Adam Baddeley. Software defined radios. *Military Technology*, 35(9):116 – 120, 2011. URL: http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=67467961&lang=pt-br&site=ehost-live.

[60] Lucian Morgos e Nistor Daniel Trip. Considerations about the modeling of software defined radio for mobile communications networks. *Journal of Electrical & Electronics Engineering*, 2(1):170 – 173, 2009. URL: http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=51005439&lang=pt-br&site=ehost-live.

[61] Walter H.W. Tuttlebee. Advances in software-defined radio. *Electronics Systems & Software*, 1(1):26, 2003. URL: http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=9816578&lang=pt-br&site=ehost-live.

[62] John McHale. Reprogrammable radios: Fpgas enable sdr applications. *Military & Aerospace Electronics*, 21(10):12 – 21, 2010. URL: http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=54526181&lang=pt-br&site=ehost-live.

[63] Chen Yu-Hsuan, Juang Jyh-Ching, Seo Jiwon, Sherman Lo, Dennis M. Akos, David S. De Lorenzo, e Per Enge. Design and implementation of real-time software radio for anti-interference gps/waas sensors. *Sensors (14248220)*, 12(10):13417 – 13440, 2012. URL: http://search.ebscohost.com/login.aspx?direct=true&db=a9h&AN=82911615&lang=pt-br&site=ehost-live.